TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Muhammad Farhan

Image Analysis and Statistical Modeling for Applications in Cytometry and Bioprocess Control

Julkaisu 1204 • Publication 1204

Tampere 2014

Tampereen teknillinen yliopisto. Julkaisu 1204
Tampere University of Technology. Publication 1204

Muhammad Farhan

Image Analysis and Statistical Modeling for Applications in Cytometry and Bioprocess Control

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB104, at Tampere University of Technology, on the 23rd of April 2014, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2014

ISBN 978-952-15-3274-0 (printed)
ISBN 978-952-15-3281-8 (PDF)
ISSN 1459-2045

Abstract

Today, signal processing has a central role in many of the advancements in systems biology. Modern signal processing is required to provide efficient computational solutions to unravel complex problems that are either arduous or impossible to obtain using conventional approaches. For example, imaging-based high-throughput experiments enable cells to be examined at even subcellular level yielding huge amount of image data. Cytometry is an integral part of such experiments and involves measurement of different cell parameters which requires extraction of quantitative experimental values from cell microscopy images. In order to do that for such large number of images, fast and accurate automated image analysis methods are required. In another example, modeling of bioprocesses and their scale-up is a challenging task where different scales have different parameters and often there are more variables than the available number of observations thus requiring special methodology.

In many biomedical cell microscopy studies, it is necessary to analyze the images at single cell or even subcellular level since owing to the heterogeneity of cell populations the population-averaged measurements are often inconclusive. Moreover, the emergence of imaging-based high-content screening experiments, especially for drug design, has put single cell analysis at the forefront since it is required to study the dynamics of single-cell gene expressions for tracking and quantification of cell phenotypic variations. The ability to perform single cell analysis depends on the accuracy of image segmentation in detecting individual cells from images. However, clumping of cells at both nuclei and cytoplasm level hinders accurate cell image segmentation. Part of this thesis work concentrates on developing accurate automated methods for segmentation of bright field as well as multichannel fluorescence microscopy images of cells with an emphasis on clump splitting so that cells are separated from each other as well as from background.

The complexity in bioprocess development and control crave for the usage of computational modeling and data analysis approaches for process optimization and scale-up. This is also asserted by the fact that obtaining *a priori* knowledge needed for the development of traditional scale-up criteria may at times be difficult. Moreover, employment

of efficient process modeling may provide the added advantage of automatic identification of influential control parameters. Determination of the values of the identified parameters and the ability to predict them at different scales help in process control and in achieving their scale-up. Bioprocess modeling and control can also benefit from single cell analysis where the latter could add a new dimension to the former once imaging-based in-line sensors allow for monitoring of key variables governing the processes.

In this thesis we exploited signal processing techniques for statistical modeling of bioprocess and its scale-up as well as for development of fully automated methods for biomedical cell microscopy image segmentation beginning from image pre-processing and initial segmentation to clump splitting and image post-processing with the goal to facilitate the high-throughput analysis. In order to highlight the contribution of this work, we present three application case studies where we applied the developed methods to solve the problems of cell image segmentation and bioprocess modeling and scale-up.

Preface

This thesis work has been carried out at the Computational Systems Biology research group of the Department of Signal Processing, Tampere University of Technology. I would like to extend my sincerest gratitude to my supervisor Prof. Olli Yli-Harja for providing me the opportunity to work in his vibrant group on such an interesting and interdisciplinary subject. It was his constant support, guidance and encouragement that helped me all the way through my thesis. I am highly indebted to all my co-supervisors Dr. Antti Niemistö, Dr. Pekka Ruusuvuori and Dr. Tommi Aho for their continuous help, guidance and advices without which this thesis work would not have been possible.

I owe a debt of gratitude to Prof. Christoph Dehio and his research group at Biozentrum Basel, Switzerland for their collaboration and also for their hospitality during my research visit to Basel. Especially, I am extremely thankful to Dr. Pauli Rämö and Mario Emmenlauer, M.Sc., for their kindness and assistance during our collaboration. I am also very thankful to Galilaeus Oy for sharing their bioprocess data with us. I would also like to thank the reviewers, Prof. Dr. Michal Kozubek and Dr. Merja Oja, for their careful evaluation of the manuscript and for their valuable comments and suggestions that helped me in improving the manuscript.

I am very grateful to all my colleagues and co-authors for their help and support throughout my studies. Especially, I would like to thank Dr. Matti Nykter, Dr. Heikki Huttunen, Sakira Hassan, M.Sc., and Antti Larjo, M.Sc., for their discussions and suggestions which helped me in achieving my research goals. I am also very grateful to all my friends, especially my Pakistani friends, more specifically the ones around me, as without their support, care and love, my life in Tampere would have been really miserable. I would also like to thank Ms. Ulla Siltaloppi, Ms. Elina Orava, Ms. Virve Larmila and other staff at the Department of Signal Processing for helping me with all the practical matters during my employment at TUT.

The financial support of Tampere Graduate School in Information Science and Engineering (TISE) and Nokia Foundation is also acknowledged.

Last but not least, the priceless love, support and encouragement from my parents, sisters and brother's family was the major source of inspiration that always kept me

motivated during tough phases of the thesis work. Thank you all for everything! Especially, for the Skype video chats, always full of prayers, advices and good wishes, through which we shared the moments of happiness and sorrow during good and bad times of the thesis.

Tampere, February 2014

Muhammad Farhan

Contents

List of Publications

I **M. Farhan**, P. Ruusuvuori. M. Emmenlauer, P. Rämö, C. Dehio, and O. Yli-Harja, "Multi-scale Gaussian representation and outline-learning based cell image segmentation," *BMC Bioinformatics*, 14(Suppl 10):S6, August 2013.

II **M. Farhan**, P. Ruusuvuori, M. Emmenlauer, P. Rämö, O. Yli-Harja, and C. Dehio, "Graph cut and image intensity-based splitting improves nuclei segmentation in high-content screening," in *Proceedings of SPIE 8655, Image Processing: Algorithms and Systems XI, 86550F*, San Francisco, USA, February 3-7, 2013, 10p.

III **M. Farhan**, O. Yli-Harja, and A. Niemistö, "An improved clump splitting method for convex objects," in *Proceedings of the 7th International Workshop on Computational Systems Biology*, Luxembourg, June 16-18, 2010, pp. 35-38.

IV **M. Farhan**, O. Yli-Harja, and A. Niemistö, "A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search," *Pattern Recognition*, vol. 46, no. 3, pp. 741-751, March 2013.

V S.S. Hassan*****, **M. Farhan***, R. Mangayil, H. Huttunen, and T. Aho, "Bioprocess data mining using regularized regression and random forests," *BMC Systems Biology*, 7(Suppl 1):S5, August 2013.

VI **M. Farhan**, A. Larjo, O. Yli-Harja, and T. Aho, "Modeling bioprocess scale-up utilizing regularized linear and logistic regression," in *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, Southampton, UK, September 22-25, 2013, pp. 1-6.

(***** corresponds to equal contribution and joint first authors)

The author of this thesis is the first author of all the publications except Publication V in which the author contributed equally and is a joint first author. As the first and corresponding author, the author was solely involved in preparation of the publications. In all the publications, apart from publicly available image data, all the real data used for image analysis and bioprocess study were obtained either from the co-authors or from the collaborating partners of the authors. The contribution of the author in these publications is as follows:

In Publications I & II, the author carried out the study, developed and implemented the methodology (except the implementation of logistic regression in Publication I and graph cut method in Publication II) and performed the computational experiments. The author was responsible for writing of the manuscript.

In Publications III & IV, the author participated in planning the study, developed and implemented the methods, generated synthetic data for experimentation, performed the experiments and wrote the manuscript.

In Publication V, the author contributed in design and analysis of the models, interpretation of the data, and made substantial contribution, along with the joint first author, in writing the manuscript.

In Publication VI, the author contributed in design of the study, developed and implemented the methodology (except the implementation of regression methods), carried out the experimental calculations and was responsible for writing the article except the parts related to the studied bioprocess.

List of Figures

Chapter 1

Introduction

The advent of imaging-based high-throughput experiments has revolutionized routine biological studies performed by bioscientists. All of a sudden, the role of automated imaging and image analysis become increasingly important in cytometry and other studies related to cell biology in order to meet the demands of high-throughput systems [1–3]. For example, genome-wide high-content screening studies for drug discovery observe heterogeneous cell characteristics and responses requiring a very large collection of samples to be imaged and studied on a cell-by-cell basis [4–8]. Cell image analysis methods typically in use for such studies are *ad hoc* based and therefore lack generalization. The ongoing advancements in bioimaging and the quest of studying and identifying gene-specific functions present more and more challenging image analysis tasks. Hence, there is a continuous need for development of new methods and algorithms delivering accurate image analysis results required for subsequent biological studies [9, 10].

High-throughput automated microscopy-based systems capture hundreds and thousands of cell populations simultaneously. These systems deliver huge amount of image data and manual analysis of such large number of images, besides being impractical, always lacks objectivity and reproducibility causing bias and inconsistency. Also, quantification of features from such huge amount of images is not possible with manual analysis [3, 11, 12]. Therefore, automatization of image analysis has become necessary in fulfilling the full potential of imaging-based research.

Image analysis for cell microscopy experiments typically involves detection, feature extraction and classification of cells and/or subcellular objects from cell microscopy images [13, 14]. Such analysis pipelines consist of a cascade of image processing modules. Image segmentation is usually considered as the fundamental module since accurate detection of objects relies on segmentation accuracy [12, 15]. Accurate automated cell image segmentation is essential in performing many cell microscopy image analysis tasks, e.g., tracking of cells and subcellular components in time series images [16–18],

quantification of cell phenotypes [7, 11], single-cell analysis in high-content screening experiments [19–21], identification and classification of cell cycle phases [22, 23].

Accurate segmentation of cells from cell microscopy images is a challenging task in many ways. Apart from the imaging aberrations, such as uneven illumination and out-of-focus regions, noise and low contrast, inhomogeneous cell/nucleus interior etc. [24, 25], challenging the separation of foreground objects from background, a considerably bigger challenge often faced in cell segmentation is the clustering of cells/nuclei forming clumps [26, 27]. Therefore, segmentation of cell microscopy images is usually carried out in two steps in which extraction of foreground from the images is followed by clump splitting to resolve the clumps into constituent objects [28, 29]. Clump splitting is one of the most challenging tasks that has widely been studied, especially in the context of cell segmentation [26, 30–33], since the fulfillment of the aim of single cell analysis rests on the accuracy of cell separation [2, 19, 21].

In bright field microscopy, though the utilization of z-stacks in cell segmentation is investigated and found to be successful [34], application of the above mentioned two-step procedure for cell segmentation has been a preferred approach [35, 36]. Whereas, in fluorescence microscopy, multiple fluorescent channels are utilized to capture cell nucleus, cell cytoplasm and subcellular component, labeled with respective protein markers which calls upon a further two-step procedure for cell segmentation in which cell nuclei segmentation is followed by cell cytoplasm segmentation [12]. More specifically, the basis for this design is that the results of nuclei segmentation is used as a context information for cytoplasm segmentation because the former is relatively easier than the latter due to often regular shape of nucleus and higher nuclei/background contrast [26, 37]. However, due to existence of clumps, segmentation of both nuclei and cytoplasm images is also carried out in the same two-step procedure [12, 38].

Besides the applications mentioned earlier, image analysis and cytometry could also find applications in bioprocesses. These processes involve microorganisms for the production of many active pharmaceutical ingredients, enzymes and fine chemicals. The difficulty arises in developing and controlling these processes due to involution of living cells [39, 40]. The key to unraveling the complexity of the underlying mechanisms of the processes is data mining which involves computational modeling and data analysis. It helps in gaining insight into the process and improving the product yield and the process reproducibility [41]. The goal in process modeling and data analysis is to identify the primary control parameters and to determine their values for process control. Image analysis could be valuable in achieving this goal since the data from imaging-based in-line sensors monitoring the primary process variables can be exploited as additional information in modeling.

Apart from challenges in process development and control, another problem linked

with bioprocesses is the involvement of economical risks and technical challenges hampering the experimentation and optimization of the processes directly at larger industrial scale. This requires the experiments to be performed and optimized in laboratories at smaller scales, such as in flasks, and then scaled up to larger scale, e.g., 1000 liter vessels which offers an even more challenging task in the process development. Traditionally, scale-up strategies revolve around constant criterion developed by transforming specific operational parameters of the processes [42, 43]. However, the use of scale-up criteria is able to provide only a partial solution as they can be used to determine the values of few parameters only while a typical bioprocess involves tens of parameters.

Scale-up can be considered as a modeling task where individual models are used to predict the values of bioprocess operational parameters in different scales. However, the problem of predicting operational parameters of bioprocesses in different scales is challenging and, due to data characteristics, it cannot be solved using conventional statistics and modeling approaches. The challenges typical of these processes are high-dimensional datasets with very small sample size (i.e. large number of parameters as compared to number of experiment samples), categorical predictor and predicted variables, varying, incompatible and incomparable parameter types with respect to processes and equipment causing missing parameter values and the non-linearity of the processes. Nevertheless, modern signal processing methods are able to provide novel solutions for developing a statistical modeling approach that has an advantage over the classic approach of identifying the effect of interaction of various parameters without using *a priori* biological information. This leads to the process modeling in such a way that the final model only contains variables that have effect on the process output.

Thesis Objectives

With the ever increasing demand for computational approach towards biomedical imaging and microscopy, new efficient automated image analysis methods and pipelines are needed. In the first part of the thesis the goal was to develop efficient automated image segmentation and clump splitting methods to facilitate subsequent biological analysis from high-throughput experiments where the aim is to enable consistent and quantitative analysis. The other main objective was to encourage the usage of automated image analysis methods in routine biological experiments which requires the methods to be as much parameter-independent as possible and at such higher level of abstraction that even a biologist can perform the study without any knowledge of image processing. Fulfillment of this objective is hindered by unavailability of open-access tools, modules and methods with proper validation data and results. This also affects reproducibility of the research besides creating an obstacle in enhancement and extensions of the methods. So the objective was the dissemination of developed methods along with the validation data by openly sharing them with the community to encourage routine use. Another

objective was the development of generalized methods. From generalization, the aim was to develop methods which are not only applicable in a specific application domain but can also be used in a wider range of application fields.

In Publication I, we present a methodology for segmenting cell cytoplasm in high content-screening experiments. The methodology is completely non-parametric, general and automated except that it incorporates supervised learning of cell outlines which requires manually drawn cell outlines for only a few training images. The implementation is publicly available with evaluation results on our images as well as on images from a publicly available high-content screening image database. In Publication II, we present a framework for segmentation of cell nuclei in high-content screening studies. The framework is totally in congruence with the desired objectives outlined above. That is, a totally automated nuclei segmentation module that can be integrated within the image analysis pipelines in a commonly used cell image analysis tool (CellProfiler 1.0) [11] is publicly available with benchmark data and obtained results. The framework is almost parameter independent and general within the scope of convexity of objects. In Publications III and IV, we present rectangular window-based, image-intensity-based and rule-based methods for automated splitting of clumps of convex objects with application towards cell microscopy images where the target is to split clumps of nuclei or cells. The methods were validated by using appropriate case studies in which the target was accurate splitting of clumps of yeast cells obtained from different sources. The methods are also publicly available with the test images and the obtained results.

In the second part of the thesis the goal was to study bioprocesses for data mining and process scale-up. Here, the first main objective was to model the process yield from the data at different experimental scales. Since the experiments at different scales have different control parameters, the idea is to incorporate data at all the scales and select such features (parameters) for model development which produce a general solution for all the scales. Therefore, the aim was to automatically identify key parameters and their interactions that affect the process outcome so that a simple and accurate model is developed using only the selected key parameters.

The second main objective in the bioprocesses study is the process scale-up from small scale to large scale such that during the scale-up the product yield remains constant. The aim was to devise statistical modeling approach instead of employing traditional criterion-based strategy in which manually defined criterion from *a priori* knowledge about the process is used for scale-up. Therefore, the aim was to identify the significant parameters from the developed product yield prediction model and to develop models for predicting the values of only those parameters at large scale using data from small scale. In Publications V and VI, we aim at achieving these objectives related to bioprocesses study. In Publication V, we investigated the use of regularized regression and random forests for bioprocess data mining where the task was to develop

a model for predicting hydrogen yield in a bioprocess. In Publication VI, we present a novel scale-up methodology based on regularized linear and logistic regression which was evaluated using a case study involving a bioprocess producing a cytotoxic compound.

Thesis outline

The rest of the thesis is organized as follows. Chapter 2 describes the cell image segmentation in bright field and fluorescence microscopy. It introduces and discusses the step-wise procedure for image segmentation i.e. pre-processing, initial segmentation, clump splitting and post-processing. Chapter 3 starts with briefly introducing the background and need for bioprocesses data mining. It further extends that with the discussion regarding bioprocess scale-up emphasizing the need of statistical approaches for obtaining it. It describes the theoretical background of statistical modeling tools and demonstrates how they are used for data modeling and scale-up modeling by presenting the methodology devised for them during the course of this thesis. In Chapter 4, we present some application case studies, which were published in the attached Publications, to highlight the contribution of the thesis, i.e., providing the application of the methods in cytometry and bioprocess control. Finally, Chapter 5 concludes the thesis by presenting the conclusions and discussion.

Chapter 2

Image segmentation for high-throughput cell microscopy

In an image analysis pipeline, image segmentation is usually the first step in which the aim is to divide the image in regions (usually in foreground and background regions) based on similarity among the pixels of one region and difference with the other [44, 45]. Usually, it is also the most important and difficult step in image analysis because it helps in object detection and the accuracy of the subsequent analysis depends upon the accuracy of object detection [14, 15]. It has been a longstanding desire, more so with the emergence of high-content screening experiments, to have fully automated image segmentation methods producing the results for a very large set of images without user intervention.

Automation of image segmentation is only possible when few assumptions can be made regarding intensity, shape, texture and other features of the objects being studied. For instance, if it is known *a priori* that the objects in an image are convex then the methods based on the assumption of convex objects would easily fulfill the desire of full automation [46]. Although, cell microscopy images sometimes allow such assumptions [31, 33] assisting automation but in certain cases a slight user intervention would rather be more helpful. For example, for high-throughput screening of microscopy images it might be more appropriate to employ active user intervention to make the segmentation algorithm learn the object shapes, intensities and other features iteratively to find the perfect segmentation [47, 48].

Segmentation of cell microscopy images is usually considered to be a challenging task, more so because most often the available general image segmentation methods do not yield the desired results [25, 47]. The challenge lies not only in separating cells from often low contrasting background but also in separating cells from each other [37, 38, 49]. The latter problem is called clumping of cells and can occur either naturally due to cells

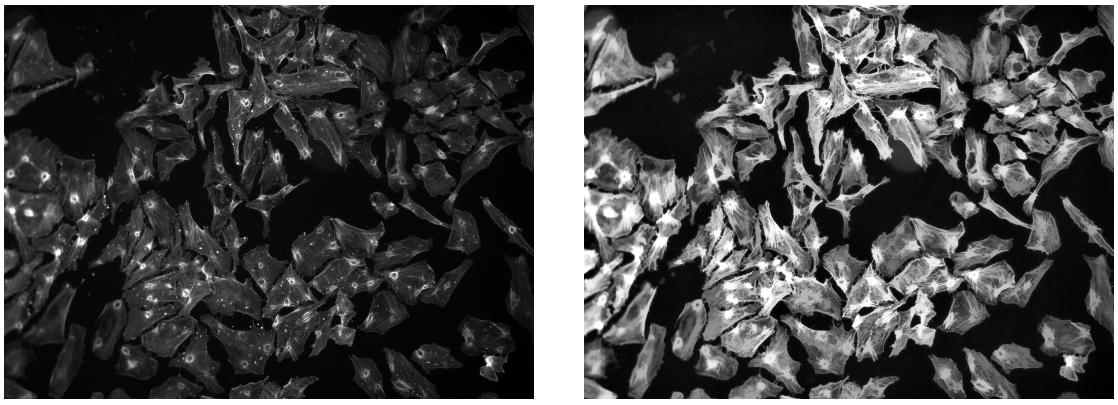
growing in buds forming clump [50], e.g. yeast cells, or due to issues related to cell culturing, optical projections [51] etc. In case of cell clumps, the average intensity values of the cells in the clumps are usually the same, also the edges are often indiscernible leading to the failure of general image segmentation methods in resolution of clumps [28]. Therefore, segmentation of cell microscopy images is generally carried out in four steps: 1) image pre-processing, 2) initial segmentation, 3) clump splitting and 4) post-processing. In some cases, post-processing is also incorporated in clump splitting whereas in other cases it is vice versa.

Despite the abundance of image segmentation methods [12, 27, 38, 52–59] there is a continuous need for improving existing methods as well as for developing new generalized, open access methods especially to meet the demands of high-throughput experiments [60]. In this Chapter, we present the methods, for performing all the four steps of segmentation, that we employed in Publications I-IV for segmentation of bright field and fluorescence microscopy images. Methods for image pre-processing and initial segmentation mainly focus on fluorescence microscopy images of cell nuclei and cell cytoplasm. Clump splitting and post-processing methods deal with both bright field microscopy images of cells and fluorescence microscopy images of cell nuclei as well as cell cytoplasms in high-content screening experiments. The usage and the utility of the methods is discussed in Chapter 4 in which we present application case studies describing how the methods are used for segmentation of cell microscopy images.

2.1 Image pre-processing

Irrespective of the imaging technique, i.e., bright field microscopy or fluorescence microscopy, used for capturing cell microscopy images the images are generally affected with noise and other imaging aberrations [22, 61]. Problems such as uneven illumination, low contrast between foreground and background, inhomogeneity among foreground as well as among background pixels, blurred and out-of-focus regions near the image corners, varying signal strengths due to improper or uneven fluorescent staining of objects and/or due to autofluorescence are typical of cell microscopy images [15]. If these problems are not taken care of, they hinder getting accurate cell image segmentation which inhibits from getting accurate results from the subsequent image analysis modules.

As far as fluorescence microscopy-based multichannel images are concerned, these problems appear in both nuclei as well as in cytoplasm channel images. However, due to higher nuclei and background contrast and due to often more regular nuclei shapes, segmentation of nuclei images presents a less challenging task than cytoplasm segmentation [26]. Nevertheless, higher segmentation accuracy generally goes hand in hand with the application of efficient image pre-processing. Therefore, the first step employed in



(a) (b)

FIGURE 2.1: Image pre-processing. (a) An actin-channel cell microscopy image showing the cell cytoplasm and (b) the result of pre-processing. The size of the image is 1040×1392 pixels.

most of the cell microscopy image analysis is pre-processing of the images to resolve the above-mentioned imaging-related issues.

In Publication I, we devised an image pre-processing technique for fluorescence microscopy images of cells in which we utilized a cascade of three image and contrast enhancement filters in order to curb most of the aforementioned problems. The pre-processing starts with the application of contrast-limited adaptive histogram equalization (CLAHE) [62] which enhances the contrast of the image. It is based on image tiling, i.e., 8×8 tiles of image are created and their respective histogram-based contrast enhancement transformation functions are computed. Each pixel's contrast enhanced intensity value is obtained using bilinear interpolation of transform functions of the four nearest tiles, based on center pixel of tiles, to avoid tile boundary artifact.

In the second stage, opening by morphological reconstruction is performed on the contrast enhanced image. Opening is performed using a marker image which is obtained by eroding the mask image (contrast enhanced image) by a flat disc-shaped structuring element of radius 5 pixels. In contrast to conventional morphological opening, this opening retains the topology of the foreground regions and helps in image smoothening and outliers removal along with solving the problem of uneven and varying fluorescent signal. In the last stage, contrast adjustment is performed on the resulting image where 1% of high and low intensity valued pixels are saturated to finally increase the foreground/background contrast. Figure 2.1 shows an original fluorescence microscopy cell cytoplasm image and its corresponding pre-processed image.

2.2 Initial segmentation

Initial segmentation generally refers to a processing step in which the aim is to separate the foreground objects, such as cells, cell nuclei, cell cytoplasms, from the background. The difficulty of the task depends on the context in which the process is to be performed. For example, in cases where objects are of more regular shape, such as yeast cells, cell nuclei in fluorescence microscopy images etc., with higher foreground/background contrast and lesser non-homogeneous object interior, the task is relatively simple and is generally performed using adaptive or global thresholding method [63–65] or gradient-based method [21]. However, there are still cases, for example, cell nuclei images in fluorescence microscopy-based high-content screening experiments, where due to noise caused by fluorescent labeling or due to other boundary variations, thresholding methods are not so useful [21, 38]. On the other hand, when the objects in the image are of irregular shape with varying signal strengths and non-homogeneous interiors, for example, images of cell cytoplasms, global or adaptive thresholding or any other binary segmentation method alone is usually found inadequate. In these cases, additional steps are needed to prepare the images such that any adaptive or global thresholding method can do the job easily and accurately [26].

In this subsection, we describe two initial segmentation methods that we used to solve the problematic cases mentioned above. The first method is used in the context of cell cytoplasm segmentation but it can be used for segmentation of regular shaped cells, nuclei as well as in other applications requiring segmentation performed in the same three/four step procedure as ours. The method employs multi-scale Gaussian representation for image enhancement so that global thresholding such as Otsu thresholding [66] method can be applied to get the initial segmentation. The second method is used in the context of cell nuclei segmentation in high-content fluorescence microscopy experiments. The method is based on widely used approach of graph cut-based image segmentation. Although the method is derived basically from [25], here we describe the main points to highlight the underlying principles.

2.2.1 Multi-scale coefficient of variation-based image segmentation

In Publication I, we presented a robust initial segmentation method mainly in the context of cell cytoplasm segmentation. However, we used the same method in successfully obtaining initial segmentation of cell nuclei images too, thus emphasizing the utility of the method. The method is based on enhancing the edges, contrast and other details of the images so that methods such as Otsu thresholding become viable. In order to assert bright field microscopy as an alternative to fluorescence microscopy, the authors in [34]

use coefficient of variation of a z-stack of bright field microscopy images to enhance the details and contrast of the image for effective image segmentation. Similarly, difference of Gaussian is a widely used technique for image edge enhancement, especially for noisy images [45]. A combination of these techniques leads to creating an image stack using multi-scale Gaussian scale-space representation of the images and utilizing its coefficient of variation image for enhancement of the low contrast foreground regions.

The method starts with creating Gaussian scale-space representation [67] of image $f(x, y)$ using

$$L(.,.;t^2) = g(.,.;t^2) * f(.,.);\ t \geq 0, \tag{2.1}$$

where t is the scale parameter and defines the width of the Gaussian kernel

$$g(x, y; t^2) = \frac{1}{(2\pi t^2)} e^{-(x^2+y^2)/2t^2},$$

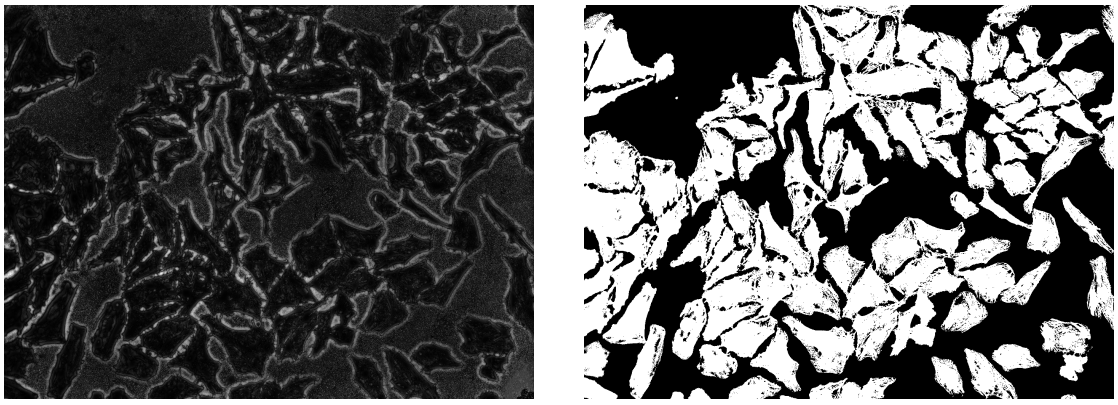
and $*$ stands for the convolution operator. The idea is to have the scale-space representation composed of seven images obtained at increasing values of t, corresponding to the original image, depending upon the magnification of the image being studied. Then, the coefficient of variation image $f_{COV}(x, y)$ is obtained by

$$f_{COV}(x, y) = \frac{\sqrt{E\left[(L(.,.;t^2) - E\left[L(.,.;t^2)\right])^2\right]}}{E\left[L(.,.;t^2) + \epsilon\right]}, \tag{2.2}$$

where $E[\cdot]$ is the expectation operator computed over the range of t for a fixed (x, y) and $\epsilon = 1$ helps in dealing with the cases of division by zero at pixels where all the scale-space images have zero intensity value. The coefficient of variation yields higher values at image background and object borders and relatively lower and consistent values at foreground pixels even in the presence of intensity inhomogeneities in the object interiors. Therefore, when the image $f_{COV}(x, y)$ is inverted and normalized and then added to the original image the result is an enhanced image $f_{enh}(x, y)$ given by

$$f_{enh}(x, y) = f(x, y) + \left(2^b - 1 - f_{COV}(x, y)\right), \tag{2.3}$$

with the intensity of the foreground pixels elevated to a relatively higher value than the background pixels, where b defines the gray-scale resolution of the image. Finally, due to the resulting increased difference between the brightest background and the darkest foreground pixels or, in other words, due to higher contrast between foreground and background pixels, intensity thresholding-based Otsu segmentation method [66] yields the desired initial segmentation. Figure 2.2 shows the coefficient of variation image of



(a) (b)

FIGURE 2.2: Initial segmentation. (a) The coefficient of variation image of scale-space representation of pre-processed image of Figure 2.1(b) and (b) the resulting initial segmentation. The size of the image is 1040×1392 pixels.

scale-space representation of pre-processed image of Figure 2.1(b) and its initial segmentation result obtained with this method.

2.2.2 Graph cut-based image segmentation

Graph cut-based minimization of energy function has widely been employed in getting the image segmentation [25, 68, 69]. This approach considers binary image segmentation as constrained labeling of each image pixel $p \in \mathcal{P}$ as either foreground or background. The constraint is defined in the form of an energy function and the minimum of that function gives the optimal labeling [68, 69]. Minimization is achieved by finding a minimum cost cut in a two-terminal graph where the pixels are assumed to be the nodes of the graph [70]. The nodes in a neighborhood are connected together with edges called n-links with their weights defining one of the two terms of the energy function to be minimized. On account of their connection with pixels in a neighborhood these weights correspond to the cost of discontinuity arising due to assignment of different labels to neighboring pixels. On the other hand, these pixel nodes are connected to two terminal nodes, source s and sink t corresponding to the two labels, and edges joining them are called t-links with their weights defining the other term of the energy function. Due to their links with terminal nodes these weights represent the cost of the wrong label assignment.

Potts model defines the energy function [69] by

$$E(L) = \sum_{p \in P} D_p\left(L_p\right) + \sum_{(p,q) \in N} V_{p,q} \cdot \delta_{(L_p \neq L_q)}, \tag{2.4}$$

where $D_p(L_p)$ and $V_{p,q}$ define the edge weights for t- and n- links, respectively and $(p,q) \in N$ is the neighborhood system. Here, Markov Random Fields (MRF) assumption is employed constraining to the consideration of only dissimilarly labeled neighboring pixel and by its Maximum a Posteriori (MAP) solution $D_p(L_p)$ is defined [68] by

$$D_p(L_p) = -\ln \Pr(I_p|L_p), \tag{2.5}$$

where $\Pr(I_p|L_p)$ denotes the probability of a pixel with intensity I_p when it belongs to label L_p. In our case of binarization, the value of $D_p(L_p)$ is set to either 0 or K based on setting the probabilities by analyzing image histogram using two thresholds for background and foreground intensities. In [68], Boykov *et al.* defined $V_{p,q}$ by

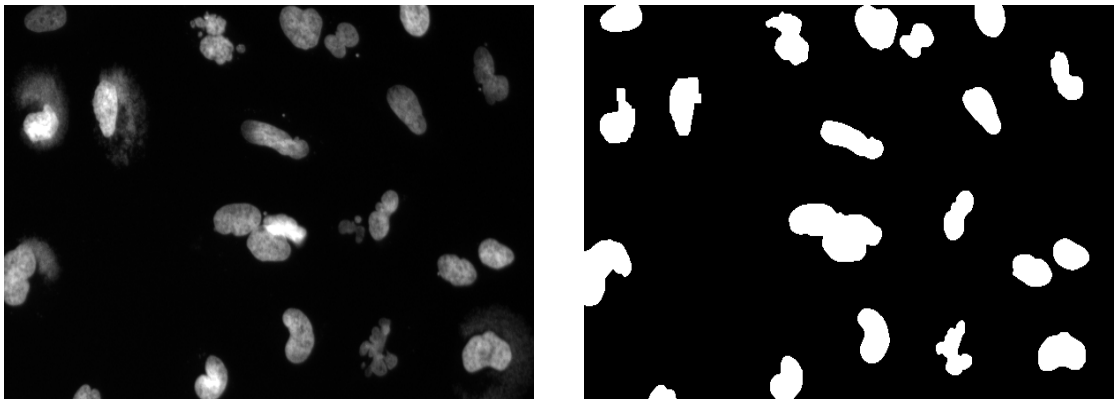
$$V_{p,q} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}, \tag{2.6}$$

where the first multiplier incorporates image gradient in which σ is the estimated average gradient magnitude in the image and $dist(.)$ is the distance metric. This definition comes naturally from the fact that the cost of assigning different labels to neighboring pixels with similar intensity value should be bigger, more so when the neighboring pixels are least distant.

The minimal cost optimal cut in the graph is an exhaustive search problem due to the presence of so many cutting possibilities and the solution in polynomial time is generally found using maximum flow algorithm with the analogy that the maximum water flow from the source to the sink is obtained through the pipes (edges) with high capacity (weights) [69]. In Publication II, we used the graph cut implementation for segmentation of cell nuclei images from [25] which used Riemmanian metric in definition of $V_{p,q}$ that was proposed in [71]. Figure 2.3 shows a pre-processed fluorescence microscopy DNA-channel image containing cell nuclei and its initial segmentation from graph cut method.

2.3 Clump splitting

Clumping of cells, cell nuclei/cytoplasms is a very common and existing problem in the analysis of biomedical cell microscopy images. Splitting of clumps of cells into individual constituent cells is of utmost importance in those analysis tasks where it is needed to have single cells separated from each other, e.g in cell tracking, cell phenotype and cycle classification etc [22, 61]. The approach that is used for clump splitting depends on the context in which it is employed. For example, the approach appropriate for the resolution of clumps of cell nuclei and regular shaped cells is to employ methods that take into account the shapes and other salient features of the individual objects or clumps



(a) (b)

FIGURE 2.3: Initial segmentation. (a) A fluorescence microscopy DNA-channel image containing cell nuclei and (b) the resulting initial segmentation. The size of the image is 365×600 pixels.

in the image [26, 31]. In contrast, when the objects are of irregular shapes, e.g. cell cytoplasms, shape- and salient features-based methods are inappropriate and a different approach such as model-based cell separation or distinguishing or detecting the borders of the touching cells needs to be employed [12, 38].

There are several methods in the literature for splitting the clumps of regular shaped objects e.g., methods based on mathematical morphology and watershed [27, 37, 72–75], ellipse fitting or shape modeling [76–81], and concavity point analysis-based methods [30–33, 46, 82–84] etc. The methods from the first two approaches are often found struggling or computationally complex when the clumps are dense and complex [28]. Therefore, enhancement or extension of available or development of completely new concavity point analysis-based methods is perhaps the appropriate direction of work. Methods available for separation of irregular shaped objects, e.g. cell cytoplasms, from each other utilize methods based on image gradient, active contour and deformable model-based methods [12, 26, 38, 47, 52]. Due to the abundance of actin filaments in cell cytoplasm, they are typically labeled and imaged and the detection of cell cytoplasm actually becomes detection of signal emitted from the protein labeling the actin filaments. However, since actin filaments are usually spread inside the cells in haphazard way, separation of clumped cytoplasms is very challenging and already available methods are found incapable in accurate separation of individual cell cytoplasm. Therefore, development of new, generalized method is needed in this case.

Concavity point analysis-based methods are quite effective and well known for splitting of clumps in cell microscopy images. The reason behind these methods being popular is that they try to imitate the human approach of separating clumped objects by looking for some prominent points, called concavity points, on the object contour and then drawing a line between those point-pairs such that a certain set of conditions

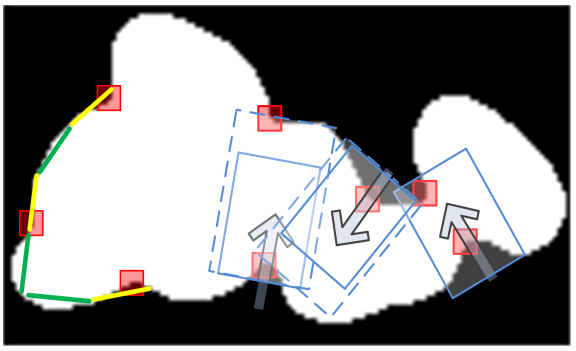


FIGURE 2.4: Concavity point detection on the left where contour segment traced by green and yellow lines are convex and non-convex respectively. Concavity point-pair search on the right where directional vector and corresponding window mask is used for finding pair of a concavity point.

is satisfied. These methods assume that the objects in the image are convex. This assumption holds very well for the target application area of biomedical cell microscopy since most cell nuclei and many types of cells have the convex shape [31, 33]. Under such assumption, any object that is non-convex besides being big enough to be classified as noise is assumed to be a clump of two or more of the constituent objects. Figure 2.4 shows a clump of multiple objects along with the concavity points.

The concavity point analysis-based methods usually involve the following three-step procedure: detecting concavity points, listing candidate split lines and choosing the best split lines or split path [33, 82]. The methods mainly differ in the approach used to implement these three steps. Apart from the deficiencies of the existing concavity point analysis-based methods in performing these three steps, there are other issues remaining to be addressed properly. For example, in case of complex and dense clumps of objects there tend to be holes within the clumps. One of the drawbacks of the existing methods is that they seldom consider prominent points on the contour of such holes for finding the split lines from them. Secondly, none of the earlier methods effectively incorporate the gray-level intensity values of the image for splitting the clumps. This also means that almost all the existing methods assume the images to be already binarized which results in the overall automated analysis being hugely dependent on the accuracy of the binarization [28]. Finally, the problem with most of the existing methods is that they do not perform a directed search for the best pair for a concavity point and consider all the other points as candidates. This creates high dependency on several user-defined parameters making it very difficult or almost impossible to find an optimized general set of parameters for a large, diversified set of images.

In order to resolve the above-mentioned issues and also to improve the results of the main steps, new methods as well as modifications to the existing methods is needed. In this subsection, we describe four methods that we used to split clumps of cell nuclei

and cell cytoplasms as discussed in our case studies in Chapter 4. The first three methods are presented in the context of splitting clumps of regular-shaped objects and are based on concavity point analysis whereas the fourth one is discussed in the context of separating clumps of cell cytoplasm. The first method is a modification of an existing method where apart from modifications, we employ the best available methods from the literature to accomplish the three steps of the methodology. The second method is a nonparametric method that incorporates the holes inside the clumps and is based on finding concavity point-pairs by using a variable-size rectangular window. Since the pairs being found already, the image intensity is not used and concavity point-pairs are joined with a straight split line. However, in the third method the original image intensity is employed in finding the path of the minimum or maximum intensity from one concavity point to the other on an accompanying gray-scale image for splitting the clumps. This results in better split accuracy with true object areas. Finally, we present a method for separating irregular shaped objects, such as cell cytoplasm in our case. The method is based on cell outline learning where a classifier is trained using spatial and transform domain image pixel-level features for classifying image pixels as cell outlines/non-outlines.

2.3.1 Rule-based method for clump splitting

In Publication III, we presented a rule-based clump splitting method for convex objects which was a modification of the method from [82] addressing the deficiencies present in it. The first step in concavity point analysis-based methods is detection of concavity points from contour of clumps. There are several concavity point detection methods in literature [30–32, 82] but curvature analysis-based method [33] is found to be a more accurate one and is therefore employed here. With such analysis, boundary points with the value of curvature above a predefined threshold are regarded as the detected concavity points, allowing multiple concavity points to belong to a single concavity region. A clump with N concavity points can have $\binom{N}{2}$ number of possible split lines. Therefore the next step is to shortlist the candidate split lines by removing invalid and intersecting lines. We used Delaunay triangulation similar to [33] for its properties of disallowing intersecting lines and of maximizing the interior angles of the triangles formed by the split lines.

Next, in order to finalize the best split lines a set of features are extracted from the images for all the concavity points involved in the candidate split lines. For candidacy, a pair of concavity points should have small distance between them besides having enough concavity depth CD associated with each of them. CD is the perpendicular distance between the concavity point and the convex hull chord of the concavity region to which

the concavity point belongs. Moreover, their respective concavity regions should be oppositely aligned. The saliency feature SA is used to test the first two conditions whereas the third condition is evaluated by two alignment features concavity-concavity CC and concavity-line CL alignments. We modified the expression for SA from [82] to minimize the existing nonlinear relationship between concavity depth and length of split lines by defining it as

$$SA_{i,j} = \frac{\min\left(CD_i, CD_j\right)}{0.1 * \min\left(CD_i, CD_j\right)^2 + D_e\left(C_i, C_j\right)^2} > 0, \tag{2.7}$$

where $D_e(C_i, C_j)$ is the Euclidean distance between the two concavity points. From this definition, concavity points with less concavity depth are allowed to have reasonably long valid split lines while discarding the quite long invalid split lines for points with large concavity depth values.

The alignment features are described by the orientation of the concavity regions which is given by a directional vector. The opposite alignment condition ideally requires the split line and the two opposite directional vectors to be aligned together. We defined the directional vector such that it bisects the concavity region in the vicinity of concavity point rather than from the convex hull chord. A local chord is obtained by joining k_{th} contour point on either side of the concavity point. The directional vector is then defined such that it has its tail at the midpoint of the local chord and head towards the concavity point. The two alignment features $CC_{i,j} \in [0, \pi]$ and $CL_{i,j} \in [0, \pi/2]$ are calculated from

$$CC_{i,j} = \pi - \cos^{-1}\left(v_i \cdot v_j\right), CL_{i,j} = \max\left(\cos^{-1}\left(v_i \cdot u_{ij}\right), \cos^{-1}\left(v_j \cdot u_{ji}\right)\right), \tag{2.8}$$

where v_i and v_j are the directional vectors of the two concavity points and u_{ij} is the vector along the line from point i to point j. It is evident that $CC_{i,j}$ determines the degree of opposite alignment of the two concavity regions whereas $CL_{i,j}$ determines the alignment of the two regions with the split line. Since both the alignment features should have minimal value, we defined the cost-function such that the point minimizing it for a particular concavity point is regarded as its best pair. The cost function is defined as

$$CF_{i,j} = SA_{i,j} + CC_{i,j} + 2 * CL_{i,j}, \tag{2.9}$$

where $CL_{i,j}$ is scaled to make up for its range being half of $CC_{i,j}$. Finally, the best pair for each concavity point is selected such that no point-pair is taken if both the points have already been used in another split line. The selected point-pairs are joined by straight lines on the already binarized image to split the clumps.

2.3.2 Variable size rectangular window-based splitting

The rule-based clump splitting method worked well for the targeted application but lacked accuracy when more complex clumps were presented to it. Therefore, in Publication IV, we presented a novel clump splitting method for splitting clumps of convex objects. The method utilized a variable-size rectangular search window to find the best pair for a concavity/prominent hole point thus resulting in reduced parameter dependency. Moreover, usage of prominent points on the contour of holes helped in achieving increased segmentation accuracy for very dense and complex clumps.

First, the method detects concave contour segments using the definition of convexity and finds the concavity point as the point with maximum curvature on that contour segment. Beginning with a starting point on the object contour, another contour point 20 contour pixels[1] apart is taken and a straight line is envisaged between them. In case of concave contour segment the line would go through the background. The value of curvature is found for every contour pixel in that segment by calculating their distances from their respective imaginary local chords, defined in Section 2.3.1, provided the midpoint of the chord is a background pixel. A fixed threshold value of 2 pixels is used to ensure the rejection of concavity points resulting due to noise and boundary irregularities. The process is repeated by traversing the contour in clockwise direction with the initial point of the next segment being either the third or twenty-third contour pixel from the initial point of the previous segment in case of convex or concave segment, respectively. The method also detects prominent points on contour of holes as points with the largest distance from their corresponding imaginary local chords such that the midpoint of the chord is a background pixel. Figure 2.4 highlights the procedure for detection of concavity points using this method.

The method then combined the usual two steps of finding candidate split lines and choosing the best ones from them by looking for only the best point-pair for every concavity/prominent hole point. We mentioned in the previous subsection that the directional vector of a concavity point describes the orientation of the concavity region. The fact that the split line is highly likely to be located in and around that region led us to use the directional vector to search for the best pair of a particular concavity/prominent hole point. From this emerged the idea of creating a rectangular window along the directional vector with its size being varied until a pairing concavity/prominent hole point is found or the upper limit of the window size is reached, see e.g. Figure 2.4. One of the two sides of the rectangular window, corresponding to window width w is obtained by extending the local chord of the concavity point from either side. The other side of the rectangular window, corresponding to window length h is obtained by using the directional vector and basic trigonometric relations. The maximum length of a split

[1]This value of 20 contour pixels is empirically obtained from multiple diversified image sets

line allowed in an image set governs the maximum window length h to be used.

Starting with a small w and comparatively large h a window is created to be used as a mask to search for the pairing concavity/prominent hole point. Small window width somehow nullifies the existence of multiple concavity/prominent hole points inside the search window, however, even if there exist more than one points inside the search window the point closest to the subject concavity point is taken. With such setting, if a pairing point is not found then the window width is increased gradually until we find one or the maximum value of width is reached. In the latter case the window length is also increased gradually until a concavity/prominent hole point is found or the maximum of the window length is also reached. After repeating the same procedure for each concavity/prominent hole point a list of point-pairs is formed which are then joined together by straight lines on the already binarized image to split the clumps.

The obtained list of point-pairs also helps in identifying the points without a pair. This is a case when a valid concavity point was lost in the detection phase. In such a case this approach leads to under-segmentation which can be rectified by iteratively finding very large objects than normal in the clump split image. The constraint for the smallest allowed object in the image is scaled to a higher value to detect such objects on which further iterations of clump splitting is performed until convergence is reached.

2.3.3 Image intensity-based splitting

Usually, in a gray scale image when the objects clump together there seems to be a slight intensity variation along the region of clump. However, when this is not the case the image intensity is not deemed informative enough to be used as an evidence for discovering the path from which the clump should split and the methods in the previous subsection are used. The problem with those methods is that they do not give the real object contour consequently they do not outline the actual individual object areas as well. Moreover, those methods are heavily dependent on the accuracy of the initial segmentation because they work on images that are already binarized. Therefore, it is of prime importance to utilize image intensity information whenever feasible.

In such cases, rather than searching for the best pair for a concavity point the appropriate way for splitting clumps is to traverse the path of minimum/maximum intensity from the subject concavity point to a point in another concavity region or to an already drawn split line. As we mentioned before that the directional vector of a concavity point describes the orientation of the concavity region to which the concavity point belongs and that the split path is more likely to be located in an around that region in the direction of the directional vector. Therefore, in Publication IV, we implemented an algorithm which finds the split path by traversing the image pixels in a directional

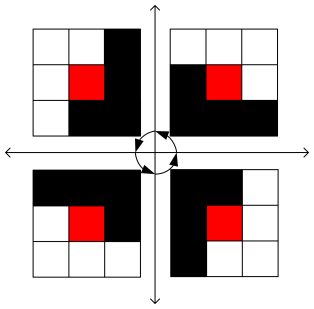


FIGURE 2.5: Four of the possible binary masks, based on the quadrant in which angle of the directional vector lies, to search for minimum/maximum split path. The black and red pixels are Don't Care, where red represents the current pixel.

search governed by the directional vector associated with the concavity point.

The search begins with the concavity point set as the current point with a 3×3 mask centered at it for locating the next lowest/highest intensity valued pixel in that particular 3×3 neighborhood of the image. Based on the angle of the directional vector associated with the concavity point there are four 3×3 masks that can be used in the directional search, see Figure 2.5. Since every concavity point has a fixed directional vector defined by the orientation of the concavity region in the vicinity of the concavity point the search mask used for a particular concavity point remains fixed as well. Once the mask is defined, the search for the next lowest/highest intensity valued pixel in the 3×3 neighborhood of the current pixel goes on as long as a background pixel is not reached in the accompanying binarized image.

The pixels belonging to the low/high intensity values in the traversed path during the directional search are assigned the background label in the binarized image provided the size of the resulting individual objects is larger than the smallest allowed object in the image, and the end point of the path belongs to another concavity region or an already drawn split line or is part of the image border.

2.3.4 Supervised learning-based outline detection for splitting

Separation of cell cytoplasms from each other is a challenging task because of irregular cytoplasm shapes and indiscernible cell boundaries [26]. Method based on detecting cell cytoplasm outlines is one of the viable solutions for this problem. However, detection of cell cytoplasm outlines is difficult as well as different in the sense that usual learning-based boundary or outline detection methods [85–88] typically detect and model distinct outlines incorporating the available shape information for detecting objects or regions in the image. In our case neither the outlines are distinct nor the shapes of the objects, and often there is a need of revealing the hidden outlines along with detecting the visible ones. Investigations revealed that the intensity and other features of the pixels with

underlying hidden outlines are very similar to those of visible outline pixels. Therefore, in Publication I, we presented a supervised learning-based method where a classifier is trained using a large set of local pixel-level image features to classify the image pixels as outlines/non-outlines for detecting the cell cytoplasm outlines in order to split their clumps.

Design of a classifier yielding high classification accuracy requires the most appropriate and informative features to be used [89]. However, assessment of such features is problematic even for a particular classification problem demanding the inclusion of general features along with the problem-specific ones in the classifier design [90]. On the other hand, high-dimensional feature set not only causes the problem of over-fitting and hindering generalization of the solution but also leads to increased classifier complexity [89, 91, 92]. Incorporating a powerful feature selection technique in the classifier design helps in addressing these issues conveniently [90].

Regularized linear regression is a technique for data modeling which gives highly sparse models paving its way to be used in classifier design such that the designed classifier uses only few features from a high-dimensional feature set thus doing automatic feature selection [93]. Least absolute shrinkage and selection operator (LASSO) [92] is one such technique which adds a penalty term, l_1-norm of coefficient vector along with a regularization parameter $\lambda > 0$, to the least squares prediction error. This shrinks the magnitude of the model coefficients towards zero as well as towards each other which leads to a sparse model with only few features corresponding to non-zero coefficients involved in modeling [93, 94]. The advantage of this technique is that the sparsity of model varies with λ and by varying it a model can be chosen with a small trade-off between accuracy and model complexity.

Using the regularized regression a classification framework is designed, i.e. sparse logistic regression classifier, which incorporates logistic function in the regularized regression for defining the class probability $p(o|\mathbf{x}_i)$ of pixel i belonging to outline as

$$p(o|\mathbf{x}_i) = \frac{1}{1 + e^{\left(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}\right)}}, \tag{2.10}$$

where o denotes the class "outline" and probability for the class "non-outline" n is given by $p(n|\mathbf{x}_i)$ = 1 - $p(o|\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbb{R}^p$ is the feature vector of the i^{th} pixel and $(\beta_0, \boldsymbol{\beta})$ is the coefficient vector. One way to estimate the coefficient vector is to maximize the penalized log-likelihood given by

$$\sum_{i=1}^{N} \left\{ \log p\left(o|\mathbf{x}_i\right) + \log\left(1 - p\left(o|\mathbf{x}_i\right)\right) \right\} - \lambda \left\|\boldsymbol{\beta}\right\|_1, \tag{2.11}$$

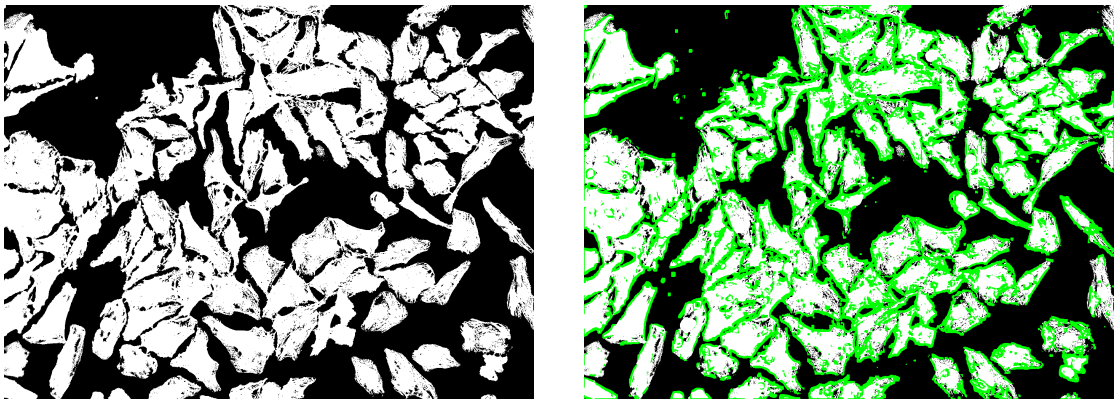
which becomes penalized iteratively re-weighted least squares problem after quadratic approximation and is solved by coordinate descent algorithm [95].

TABLE 2.1: Filtering operations and the filter parameters for computing pixel-level features from training images.

Operation (Feature)	Parameter	Values	Total
Gaussian low pass	kernel width σ	3:2:49	24
Integrated pixel int.	kernel size	3:2:9	04
Laplacian of Gaussian	kernel width σ	3:2:49	24
Difference of Gaussian	kernel width σ		05
Morphological top-hat	kernel size	3:2:49	24
Morph. bottom-hat	kernel size	3:2:49	24
Local binary pattern and contrast	(quantization, radius)	(8,1)	02
Variance	kernel size	3:2:49	24
Order statistics (Min., Med., Max.)	kernel size	3:2:7	09
Haralick (13-features)	kernel size	5:2:15	78
Gabor filter	kernel size, freq. f, orientation θ	5:2:15, 1/4:1/4:3/4, $0{:}\pi/4{:}3\pi/4$	72
Total number of features.			290

Exploiting the capability of sparse logistic regression in giving highly sparse models, we purposefully created a high-dimensional feature vector from training images including such generic linear and non-linear features which are deemed useful in our desired methodology. In this way the designed framework becomes general enough to be used for other similar classification problems. Moreover, computational overhead is reduced because calculation of such large feature vectors is only limited to training image(s) and only a very small number of selected features are required to be calculated from test images.

For training the classifier, manually created cell cytoplasm outlines produced by biologists are used. Since the classifier used is capable of dealing with $P \gg N$ cases[95, 96] so the pixels of only one image are more than enough to train the classifier and the rest of the images in the dataset can be used for evaluating the performance of the classifier. This makes unbiased evaluation possible for even very small benchmarking dataset. The training image(s) are input to a large filter bank, comprising spatial and transform domain filters with varying parameters, to extract local pixel-level features. The feature set comprises general intensity-based, edge features and textural features such as local binary pattern (LBP) [97], Gabor features [98] with varying scale and orientation and Haralick features [99]. A set of 290 features are calculated for each pixel in the training image using filters with varying kernel sizes, see Table 2.1 for set of features extracted from training images. A total of 1000 pixels, 500 outline and 500 non-outline, are picked at random from the training image and the corresponding 1000×290 feature vector and 1000×1 target labels are provided to the sparse logistic regression classifier. 10-fold cross-validation is performed to estimate the prediction errors for different models obtained with varying values of λ and the model yielding the minimum prediction error



(a) (b)

FIGURE 2.6: Outline/Non-outline detection. (a) An image after initial segmentation. (b) Resulting outlines (green) from classification of image pixels into outline/non-outline pixels. The size of the image is 1040×1392 pixels.

or the one with error within one standard error of the mean cross-validation error is selected. For the test image, the selected features are extracted and input to the classifier which gave posterior probability values of the image pixels which, in ideal case, gives the class label (outline/non-outline) for every pixel but generally needs some post-processing for delivering the desired accuracy. Figure 2.6 shows the result of outline detection for a fluorescence microscopy cell cytoplasm image where it is clear that refinement is needed through an additional step of post-processing.

2.4 Post-processing

As we mentioned earlier, sometimes post-processing and clump splitting might be interchangeable terms while in some cases they might be separate stages of the overall segmentation methodology. However, apart from the need of clump splitting there can be other processing steps required to improve the overall accuracy of segmentation methods. For instance, region merging is a complementary step to clump splitting which makes up for over-splitting caused by clump splitting method [14, 15]. Also, the shortcomings of the initial segmentation are dealt with in the post-processing phase. Moreover, post-processing of the classifier outputs is generally a complementary part of any classification framework [89]. Except for the concavity point analysis-based clump splitting method incorporating image intensity, all the other three clump splitting methods discussed above require another step of post-processing to further refine the clump splitting results for achieving higher segmentation accuracy. The post-processing techniques also depend on the context in which they are applied. For example, for images containing objects with regular shapes, post-processing typically utilize the prior shape

information, whereas for images of irregular shaped objects other context-based technique using methods from mathematical morphology might be more appropriate.

In Publication IV, we devised a post-processing technique which can be applied to the methods of Section 2.3.1 and 2.3.2 for removing residual objects and objects not in accordance with the prior shape information as well as for removing non-smooth contours caused by formation of acute angles between split lines. The origin of all these problems is the usage of straight lines for joining concavity-point pairs. First, the list of concavity point-pairs is checked to find points involved in two lines, i.e. have degree equal to two. In case of concavity point with degree two, other two points of the two lines are checked if they have no other non-shared split line. In a positive case centroid of the triangle between the three points is found and all the three points are joined with the centroid to get smoother object contours. Second, concavity points with degree three are found in the list that are creating two very acute angles between split lines. In this case, the pair of lines with the narrowest angle is found and their other two concavity points are examined to have either only one or both of them with degree equal to two. In the former case the split line involving that particular concavity point whereas in the latter case the line involved in the wider of the two angles is discarded from the list. When neither of those two concavity points have degree two then the normal post-processing is performed one after the other for the two pair of lines.

The outputs from the classifier designed in Section 2.3.4 are class probabilities and a threshold probability value of 0.5 gives thick outlines due to highly matching features in the pixels nearby outlines. Also varying signal strength, noise and intensity inhomogeneities inside the cell result in discontinuous outlines as well as pixels interior to cell cytoplasm classified as outlines. Therefore, post-processing is required to refine the classifier output for accurate segregation of cell cytoplasms. In Publication I, we exploited the DNA-channel cell nuclei images to be used as contextual information for indication of cell as well as cell outline locations.

First, using segmented nuclei pixels as the evidence of cell interior, pixels inside the cell interior which are misclassified as outline pixels are filtered. The nuclei images also help in refinement of initial segmentation where the small holes caused by intensity inhomogeneities are filled out. Complement of this image help in further strengthening the outlines when combined with the filtered outline image. Second, morphological skeletonization is performed for thinning the outlines. It gives single-pixel cytoplasm outlines along with non-connected branches caused by noise and/or discontinuous outlines. For closed outline contours some of these non-connected branches need to be joined based on the fact that most often only one nucleus corresponds to one cytoplasmic region. Moreover, due to these branches and false detections in thinned outlines, over- and under-splitting occurs and an additional step of splitting and merging is required.

In order to find the object correspondence, first-stage cytoplasm segmentation is

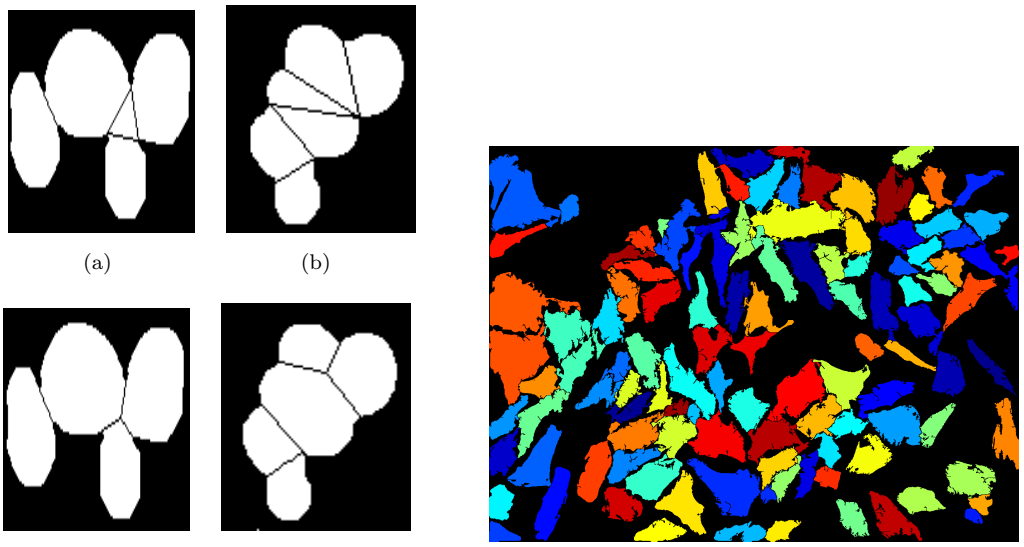


FIGURE 2.7: Image post-processing. (a)-(b) Two problematic cases of concavity point analysis-based method and (c)-(d) the images after post-processing. (e) Post-processing result for the outline/non-outline detection of Figure 2.6.

obtained by applying thinned outlines on initial segmentation. It is then morphologically reconstructed using nuclei image which helped in getting one-to-many (requiring splitting), many-to-one (requiring merging), and one-to-one correspondence between cytoplasm and nuclei. In case of splitting, morphological closing is performed to smoothen inside of cytoplasm and to extract branches which are then dilated to bridge the gaps. The thickened outlines are again thinned using the same procedure. The advantage of not forcing the splitting of a cytoplasm to get one-to-one correspondence when the outlines were not there is that over-splitting affected by nuclei over-segmentation is avoided and also the multi-nuclear cell phenotypes get retained. For merging, the to-be-merged regions are dilated to find the overlapping regions in nucleus-bearing candidate cytoplasmic regions. Solidity of the resulting regions is used to select the most appropriate candidate in case of multiple candidates. Two more iterations of the procedure helps in merging the distant to-be-merged regions as their adjacent such regions get merged in the process. Finally, h-connectivity and 8-connectivity of the objects are removed and the small holes are filled using basic operations from mathematical morphology. Figure 2.7 shows the results of post-processing for the two cases of clumps of regular and irregular shaped cells.

2.5 Validation and performance evaluation

Validation of image segmentation methods before using them as tools in routine biological analysis is necessary because image analysis pipelines involving subsequent analysis depend on the performance of image segmentation. Previously, validation was typically

performed by qualitative evaluation of the result by visual inspection. However, qualitative evaluation besides being subjective and biased is not anymore feasible especially with the emergence of high-throughput experiments involving hundreds of thousands of images being studied simultaneously [10, 100]. The other traditional way of validation is based on manual analysis where the results of the methods are compared against manually annotated or manually created ground truth images [13, 14]. Although it also has the same drawbacks along with issues such as multiple ground truths at pixel-level due to multiple annotators but it produces quantitative and consistent results. Moreover, the automation of the process keeps the bias limited to the creation of ground truth. One way of removing this bias too and moving towards completely objective measures is the development and usage of simulation-based validation in which synthetic microscopy images generated with realistic characteristics are used along with generated ground truth information for validation [101]. Sometimes generation of synthetic microscopy images matching the desired characteristics might be difficult or even impossible with the already available tools, therefore we need to revert to manual analysis-based validation where indirect comparison [102], i.e. comparing results of different methods, of the analysis results assists in getting better understanding of the performance of the developed method. In all of Publications I-IV, we used manual analysis-based validation, whereas in Publication IV we also used simulation-based validation to demonstrate the robustness of our methods.

In order to evaluate the performance of our segmentation methods and to get quantitative results from validation as well as from indirect comparison, we calculated the performance metrics at two different levels: object-level and pixel-level. For object-level comparison the ground truth was created by marking the presence of an object using a marker of relatively small size as compared to the size of the object. In this case the resulting image is compared against the marker image to obtain true and false object detection measures. Whereas for pixel-level comparison the ground truth was created by drawing the outlines on the object contours and each and every pixel of the resulting image is compared with the pixels of ground truth image [13]. Hence the margin of error in exact quantification is more in object-level measures as compared to pixel-level measure since in the former case those markers are used to detect only the presence of the object in the resulting image.

The performance metric that we used throughout this study to assess the segmentation accuracy is F-measure (FM) [103] given by

$$FM = \frac{2}{\left(\frac{1}{PR} + \frac{1}{RC}\right)}, \tag{2.12}$$

which is the harmonic mean of two other measures, Precision (PR) and Recall (RC) defined as

$$PR = \frac{TP}{(TP + FP)} \quad \text{and} \quad RC = \frac{TP}{(TP + FN)}, \tag{2.13}$$

where TP, FP and FN are the primary measures for detection [103] and are termed as true positive (object was in the ground truth and detected), false positive (object was not in the ground truth but detected) and false negative (object was in the ground truth but not detected), respectively. Lower values of FP and FN govern higher values of PR and RC which amounts to a higher value of FM and thus the segmentation accuracy.

Where object-level measures just give the object count and the number of falsely detected objects, pixel-level measures, on the other hand, provide the insight into a more real evaluation of the segmentation accuracy. That is, it helps in getting the real object-level measures by finding the correspondence between the objects in the resulting and ground truth images using the amount of overlap in terms of pixels. For each object in the ground truth image the object with maximum overlap in the segmentation result is extracted and pixel-level performance metric is calculated. A threshold value of $FM_{th} = 0.6$ ascertains the correspondence and the corresponding object is removed from the segmentation result for finding the correspondence of the next ground truth objects. Hence one-to-one (TP), one-to-none (FN) or none-to-one (FP) object-level correspondence is obtained between the ground truth and the segmented image where TP gives the object count and the other two quantify the false detections.

Chapter 3

Bioprocess data mining and scale-up

In bioprocesses, living microorganisms are used to produce bioproducts such as active pharmaceutical ingredients, enzymes, biofuels and fine chemicals. The problem with these processes is that they are difficult to develop and control because of the usage of living cells [39, 40]. Also, these processes typically involve many operational and control parameters which increase the complexity of the problem thus requiring data mining for analyzing and modeling the process and its production properties. For example, identification of the primary control parameters affecting the product yield and determination of their values for controlling and optimizing the process is necessary in modeling and data analysis so that the product yield can be maximized [43, 104]. This is similar to the task of selecting the most important features to build a general model for data analysis. However, developing general model is problematic due to data characteristics and requires computational statistical modeling approach to be used.

Due to the considerably costly large industrial scale production, bioprocess development is usually carried out at small scale such as in flasks. Once the bioprocess is optimized it is scaled up to large scale which is challenging besides carrying economical risks [43, 105]. This is because the prediction of operational parameters in large scale fermentation using small scale data is complicated and, due to data characteristics, conventional statistics and modeling approaches are usually unsuitable. The existing approach of developing scale-up criteria [39, 40, 42, 43, 104–107] make it possible to determine the values of particular operational parameters in different scales. However, they are not capable enough to determine values of more than just a few parameters, while a typical bioprocess involves tens of parameters. Moreover, they require manual identification of the most important parameters to build a scale-up criterion which requires

a priori knowledge of the process. On the other hand, statistical modeling-based approach [108–110], such as response surface methodology (RSM)-based methods, though help in optimization as well as in studying interactions between several variables, but they cannot be used to predict the values of operational parameters in large scale based on the samples in small scale.

Nevertheless, a statistical modeling approach can be devised which models a process enabling the prediction of its operational parameter values at different scales. This approach would also reveal the interactions between various parameters and their effects on the process outcome leading to such process modeling that the final model only contains variables that have effect on the process output. Thus a rather automatic selection of the most important parameters for the process development is performed. However, again due to data characteristics, such as very few high-dimensional samples, categorical parameters, non-linearity of the process and missing values due to different equipments and parameters at different scales etc., process modeling is non-trivial demanding a computational statistical approach.

In this Chapter we present the methods for modeling bioprocess that we employed in Publications V-VI to predict the product yield at different scales as well as to achieve the process scale-up. In Publication V, we explored two different methods, i.e. regularized linear regression and random forests, for characterizing the behavior of a bioprocess under specific conditions. In Publication VI, we developed computational methodology for bioprocess scale-up which is based on process modeling for predicting the values of bioprocess operational parameters in different scales. Product yield correspondence is employed in the scale-up modeling, i.e., the product yield of the predicted large scale sample and given small scale sample are roughly the same. Our approach exploits embedded feature selection property of regularized linear and logistic regression for modeling of process yield and prediction of large scale parameter values using small scale data, cross-validation for model selection and logistic regression for analysis and classification of categorical parameter values.

3.1 Bioprocess modeling and data mining

3.1.1 Multiple linear regression

Given the modeling task where we have a response variable, e.g., product yield, and a set of predictor variables corresponding to the bioprocess operational parameters, the basic modeling technique is to use multiple linear regression model [93, 111] given by

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \ldots + b_p x_{ip}, \tag{3.1}$$

equivalently written in matrix form for $i = 1, 2, 3, \ldots, N$ as

$$\mathbf{y} = \mathbf{H}\boldsymbol{\theta} = [\mathbf{1X}]\boldsymbol{\theta}, \tag{3.2}$$

where $\mathbf{y} = [y_1 y_2 \ldots y_N]^T$ is the response variable, $\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_N]^T; \mathbf{x}_i = [x_{i1} x_{i2} \ldots x_{ip}]^T$ is the predictor vector, and $\boldsymbol{\theta} = [b_0 b_1 b_2 \ldots b_p]^T$ is coefficient vector which is estimated such that the sum of squared error in the observed and predicted values of y_i is minimized.

This simple linear model is, however, unable to cope with the non-linearity of the data because of utilizing linear combination of just the actual predictors. One way to incorporate non-linearity of data into modeling while keeping the model linear and easily interpretable is to include non-linearly transformed predictors with quadratic terms and interaction of variables. This, however, increases the number of variables consequently leading to a high-dimensional predictor vector which tends to produce over-fitting models and thus inhibit generalization [91, 92]. Also prediction vector augmented in this way usually suffers from redundancy and high correlation which, for small sample size, typical of bioprocess data, leads to the problems of rank deficiency and multicollinearity creating difficulties in parameter estimation as well as in achieving higher prediction accuracy [91].

3.1.2 Regularized regression

Regularized regression is found to be a very effective tool in handling such situations [93, 112]. It produces sparse solutions by shrinking the regression coefficients towards zero as well as towards each other [93]. By doing this it automatically performs feature selection. Moreover, it exploits the trade-off between variance and bias, i.e., decreases variance at the cost of small increase in bias, towards achieving generalization [92]. LASSO [92] is one of the methods used in regularized regression which we already described in Section 2.3.4 because of its usage in our supervised learning-based method for image pixel classification for cytoplasm outline detection. Using LASSO, the estimation of the regression coefficients in (3.1) is obtained by

$$\hat{\boldsymbol{\theta}} = \arg\min_{\theta} \sum_{i=1}^{N} (y_i - b_0 - \sum_{j=1}^{p} x_{ij} b_j)^2 \tag{3.3}$$

$$\text{subject to} \sum_{j=1}^{p} |b_j| \leq t,$$

which by using (3.1) and (3.2) is equivalent to minimizing the prediction error-based Lagrange function given by

$$\|\mathbf{y} - \mathbf{H}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 , \tag{3.4}$$

where $\mathbf{H} \in \mathbb{R}^{N \times P+1}$ assuming x_{ij} is standardized, i.e., it has zero mean and unit norm, $\mathbf{y} \in \mathbb{R}^+$, $\lambda > 0$ is regularization parameter controlling the sparsity of the model (at some large value of λ, all coefficients are zero) and $\|\boldsymbol{\theta}\|_1$ is the l_1-norm of the coefficient vector by which the error function is penalized. *Least Angle Regression* algorithm [93, 113] is used to solve for $\hat{\boldsymbol{\theta}}$ which gives a set of solutions for varying λ. Cross-validation is performed to estimate the prediction errors and the solution with the minimum error is selected.

3.1.3 Random Forests

The other method we used to model the process yield is based on decision tree where the prediction of the output values is performed in a top-down hierarchical structure called decision tree. The hierarchy is created by a set of combinatorial logic rules of if-then for comparing the parameter values or features with a threshold value for predicting the output [89]. Intuitively the prediction of output is performed via a sequence of comparisons beginning from the root node of the tree yielding subsequent nodes linked with branches corresponding to different possibilities. The creation of descendant nodes goes on before arriving at leaf node which corresponds to a particular output value. One of the main issues in decision trees is to decide when to stop growing a tree further because stopping it early will cause higher training error whereas over-fitting is caused otherwise [89]. Classification and regression tree (CART) [89, 114] is a widely used framework for constructing decision trees. It describes when to stop further splitting of a node so as to declare it to be a leaf besides helping in formation of an optimal tree by removing redundant portions and making too large trees into smaller and simpler ones [89].

Decision trees, in general, are affected by over-fitting problem when the training feature vector is small as well as high-dimensional, typical of bioprocesses. Random forests [115] provides a solution to this problem in which a large number of regression trees are constructed and the outputs of all the trees are averaged to predict the final output. We exploited the availability of a fast implementation of random forests, i.e. Random Forest with Artificial Ensembles (RF-ACE) by our colleague in [116] where we chose the number of trees in the forest, and the fraction of randomly drawn features per node split to be 20, and 10, respectively. Using its feature ranking characteristic based on statistical significance of features, we first selected a set of significant features

from the experimental data and then constructed a model for prediction of the output variable.

3.2 Regularized linear and logistic regression-based scale-up

The process modeling described in the previous section was presented in the context of modeling process output, e.g., to predict the process yield, when the process operational parameters hold numerical values. However, in practice, bioprocesses often contain process parameters holding categorical values. Although CART is capable of handling categorical predictors but linear regression requires some transformation of the predictors to use them into modeling. Therefore, in order to model the process scale-up, which also involves development of a model for predicting the product yield, we first perform encoding of categorical variables into a form that can be incorporated into the standard regression model.

3.2.1 Encoding of categorical variables

Dummy coding is a commonly known procedure for encoding a categorical variable with k labels into $k-1$ dichotomous variables (variables with two labels) [111] that can be used directly in a regression model. A coding system is desired which besides minimizing the correlation and/or linear dependency among the variables also highlights the desired comparison among the different labels. Hence, we performed dummy coding by employing the method called contrast coding which creates contrast among a set of labels by giving the same variable positive and negative values for the labels between which the contrast is meant to be created. Moreover, with such coding even if the labels for different variables remain unchanged for some samples, the values held by the subsequently created dummy coded variables would be different thus minimizing the singularity issues confronted otherwise.

3.2.2 Product yield modeling and data rearrangement

A model is developed using regularized linear regression, as described in Section 3.1, for predicting the product yield using the data obtained after dummy coding the categorical variables. As we mentioned before, the aim in the scale-up is to predict the values of process operational parameters at large scale using the small scale samples such that the product yield for the resulting large scale sample is approximately the same as the

product yield of the small scale sample. Therefore, using the product yield as reference a correspondence-based data rearrangement is performed in the next step. For one-to-one correspondence between small and large scale samples their product yield values need to lie in a specific range, i.e., a tolerance range. For each large scale sample, corresponding small scale sample(s) are found and since one-to-many correspondence is possible so large scale samples are replicated to match the number of small scale samples corresponding to it so that the final sample size at both scales remain the same.

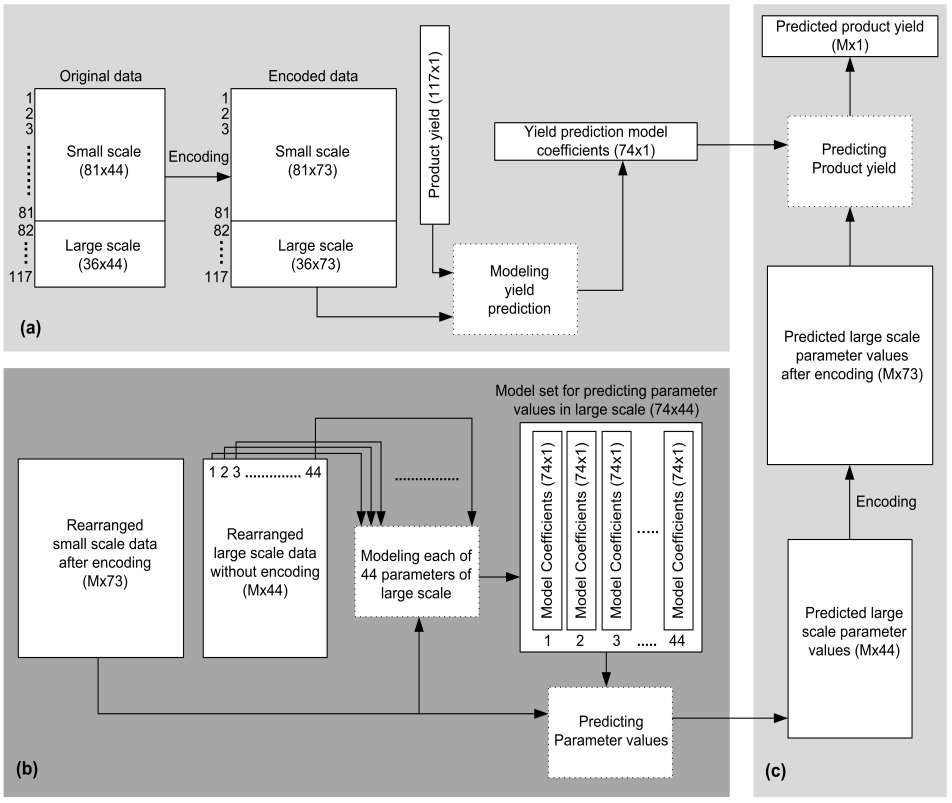


FIGURE 3.1: Methodology for scale-up modeling and testing. Procedure for (a) modeling yield prediction in all scales, (b) modeling operational parameter values in large scale using the data at a smaller scale and (c) testing the developed models.

3.2.3 Scale-up modeling

Finally a scheme for scale-up modeling is devised in which respective models are created for each numerical variable using the same concept as in Section 3.1, whereas the prediction of categorical variables are handled in a way similar to the one we used for image pixel classification in Section 2.3.4 with an exception that the variables in this case can have multiple category labels as opposed to two-category case discussed earlier. The idea in scale-up modeling is to predict the values of each variable in the large scale given the variables of small scale and the yields at both scales. Using the definition of

standard regression, the problem statement defined for the scale-up is to

$$\text{find } \hat{\boldsymbol{\theta}}_j : \mathbf{x}_{Lj} = [\mathbf{1}\mathbf{X}_S]\boldsymbol{\theta}_j \text{ given that } |\mathbf{y}_L - \mathbf{y}_S| \leq \epsilon \tag{3.5}$$

where $j = 1, 2, \ldots, p$ and which results in

$$\hat{\mathbf{X}}_L = [\hat{\mathbf{x}}_{L1}\hat{\mathbf{x}}_{L2}\ldots\hat{\mathbf{x}}_{Lp}], \text{ such that}$$

$$\hat{\mathbf{y}}_L = [\mathbf{1}\hat{\mathbf{X}}_L]\hat{\boldsymbol{\theta}}^{yieldprediction} \approx \mathbf{y}_L, \tag{3.6}$$

where the subscripts L and S are used for large and small scale data respectively, ϵ is the parameter defining the tolerance range for the difference in small and large scale product yields and can be chosen arbitrarily and $\hat{\boldsymbol{\theta}}^{yieldprediction}$ is the coefficient vector for the yield prediction model obtained using the method from Section 3.1 utilizing data from all the scales.

In Section 2.3.4, we utilized regularized logistic regression [95, 96] for a binary classification problem. The same framework is employed here for multiclass classification problem where the PDF for the class $k = 1, 2, ..., K$ is modeled as

$$p_k(\mathbf{x}) = \exp(\boldsymbol{\theta}_k^T\mathbf{x})/(1 + \sum_{j=1}^{K}\exp(\boldsymbol{\theta}_j^T\mathbf{x})), fork \neq K, \tag{3.7}$$

$$\text{and } p_K(\mathbf{x}) = 1/(1 + \sum_{j=1}^{K}\exp(\boldsymbol{\theta}_j^T\mathbf{x})), \tag{3.8}$$

where $\mathbf{x} = [1x_1x_2...x_p]^T$ denotes the augmented predictor vector and $\boldsymbol{\theta}_k = [b_{k0}b_{k1}b_{k2}\ldots b_{kp}]^T$ are k set of coefficients of the models, one for each of the k categorical label, and are obtained by maximizing the penalized log-likelihood which, for multiclass problems, is given in the form of

$$\hat{\boldsymbol{\theta}}_{1,2,\ldots,K} = \underset{\boldsymbol{\theta}_{1,2,\ldots,K}}{\arg\max}[\frac{1}{N}\sum_{i=1}^{N}\log p(x_i) - \lambda\sum_{j=1}^{K}\|\boldsymbol{\theta}_j\|_1], \tag{3.9}$$

where $\boldsymbol{\theta}_{1,2,\ldots,K} \in \mathbb{R}^{(p+1)\times K}$ and which is again solved by coordinate descent algorithm [95]. Again, 10-fold cross-validation is used to estimate the prediction errors for different models obtained with varying values of λ and the model yielding the minimum prediction error or the one with error within one standard error of the mean cross-validation error is selected.

After developing the models for the variables selected in product yield modeling, small scale samples are used to predict the large scale sample values. For prediction

of numerical variables, the estimated model coefficients and the predictor vector, i.e. small scale data, is used following (3.2) or (3.5). For categorical variables, using the coefficients of the selected model along with the predictor vector the probability densities are calculated for every class labels using (3.7) and (3.8) and the class with the highest probability is the predicted class label for the given categorical variable. The block diagram in Figure 3.1 outlines the described scale-up methodology.

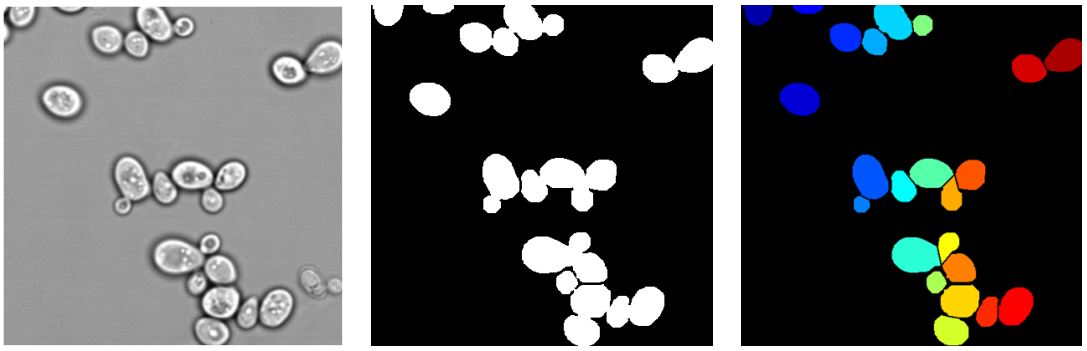
Chapter 4

Application case studies

In order to investigate and thus demonstrate the performance of the developed methods and also to highlight the contributions of this study we present three application case studies that were published in the attached Publications. However, as we already mentioned that one of the goals of this study was to develop methods and frameworks that are general enough to be used in a wide range of applications, therefore, it should be explicitly mentioned that the developed methods are by no means limited to these applications. Since the main goal of the thesis was to develop methods solving the problems related to image analysis as well as in bioprocess data mining and scale-up, we do not focus on describing the biological background nor do we discuss the subsequent data analysis that may be performed for reaching biological conclusions. Next we discuss the cases and describe how we used the methods, presented in Chapters 2 and 3, or their combination for solving them. In both the cases of image segmentation the details about image acquisition and creation of benchmark set can be found from the attached Publications.

4.1 Segmentation of budding yeast cell images

The budding yeast *Saccharomyces cerevisiae* provides a very good test case for evaluating the performance of segmentation and clump splitting methods for regular shaped objects because of their roundish and convex shape along with their natural tendency to form clumps through budding. Also, from the application point of view, accurate segmentation and separation of individual cells from clumps of yeast cells is necessary in many applications based on single cell analysis [35, 36]. Moreover, the availability of budding yeast *S. cerevisiae* cell images from the Saccharomyces Cerevisiae Morphological Database (SCMD) [117] led to its usage in testing the developed methods.

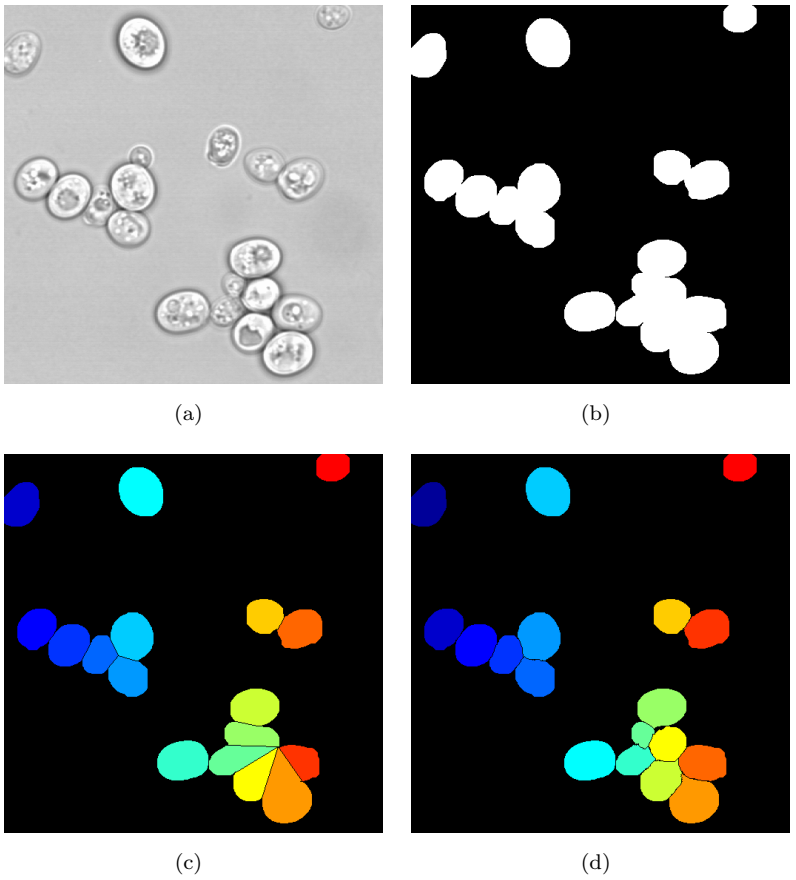


(a) (b) (c)

FIGURE 4.1: Segmentation of budding yeast cell images. (a) A bright field microscopy image of budding yeast *S. cerevisiae* cell population, (b) its initial segmentation and (c) its final segmentation result using clump splitting method of Section 2.3.1. The size of the image is 400×400 pixels.

In Publication III, we used images of yeast cells from SCMD [117] for performance evaluation. Initial segmentation of the images is performed using Otsu thresholding method [66]. Then the method for clump splitting described in Section 2.3.1 is applied on the binary segmentation result obtained in the previous step. A bright field microscopy image of budding yeast *S. cerevisiae* cell population and its segmentation result is shown in Figure 4.1. Although a better initial segmentation technique for this image would tackle with most of the clumped cells but the intention was to test the robustness of the clump splitting method because in a tougher scenario we would get a somewhat similar binarized result insisting the need for better clump splitting methods.

In order to test the clump splitting methods presented in Sections 2.3.2 and 2.3.3, in Publication IV once again we used budding yeast cell fluorescence microscopy images from SCMD. In order to manifest the generalization of our method, in fact rather more specifically, to further highlight the enhanced performance of our intensity-based method we also used bright field microscopy images of budding yeast *S. cerevisiae* from our collaborators. After image acquisition, the initial segmentation of the bright field images is performed using the method from [35]. Then, the clump splitting methods of Sections 2.3.2 and 2.3.3 are applied on the two sets of images with the constraint for the size of the smallest allowed object in the image put to 300 pixels and 50 pixels for bright field and fluorescence microscopy images, respectively. In the case of not using image intensity for clump splitting, post-processing method for regular shaped objects described in Section 2.4 is applied to make the final results resembling more to the ones manually obtained from an expert. Figures 4.2 and 4.3 present one representative case of each image type with the results obtained from both the methods. It is clear that the method based on image intensity is able to split the complex clumps as well as the clumps near the image borders much better with more accurate object areas than the



(a) (b)

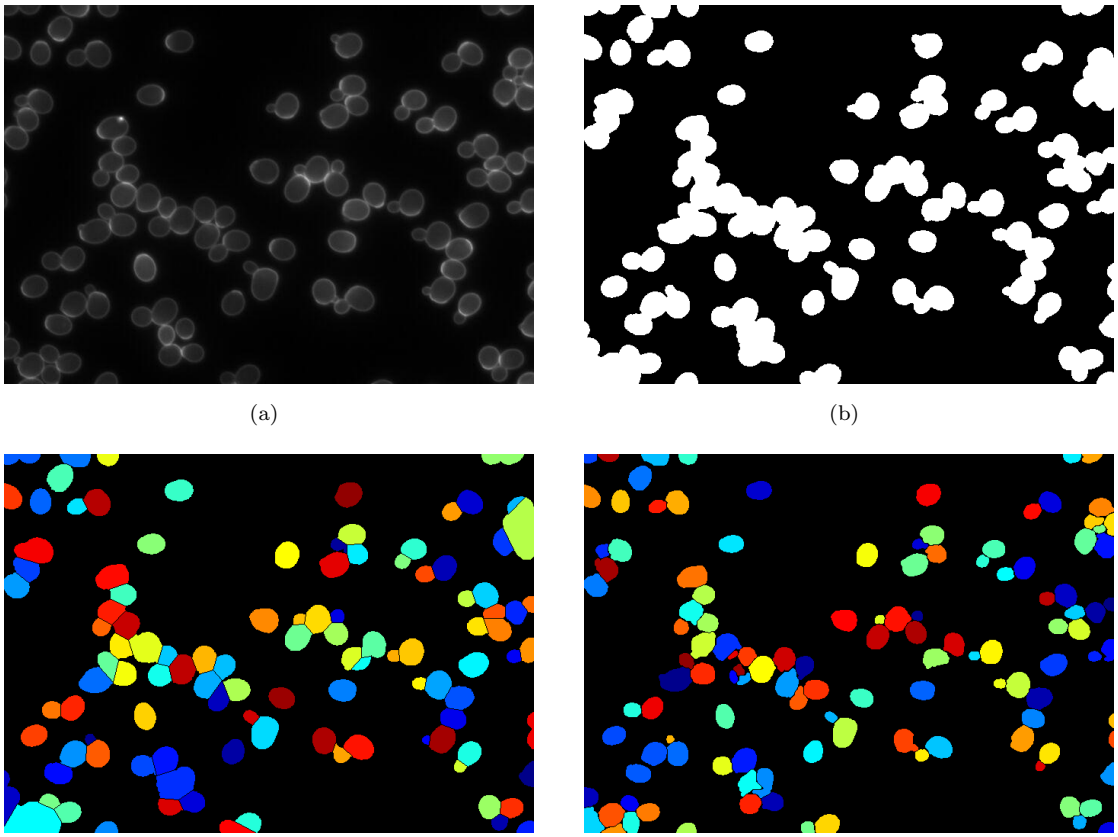
(c) (d)

FIGURE 4.2: Segmentation of budding yeast cell images. (a) A bright field microscopy image of budding yeast *S. cerevisiae* cell population, (b) its initial segmentation using the method from [35]. (c) and (d) Final segmentation results using clump splitting methods from Sections 2.3.2 and 2.3.3, respectively. The size of the image is 512×512 pixels.

method not using image intensity.

Finally, in order to demonstrate the generalization of the methods for other applications containing images of convex objects as well as for quantitative evaluation of the methods we created extremely large and diversified benchmark image sets with ground truth using the synthetic images of cell populations with realistic properties generated using SIMCEP simulation tool [101]. Here we only had binary images to work with so we applied the clump splitting method of Section 2.3.2 with the constraint for the size of the smallest allowed object in the image put to 700 pixels. Figure 4.4 shows an image from this test set which manifests the robustness of our method for splitting complex clumps, with varying object sizes as well as probability and amount of overlap, even though the intensity information is not available to produce even better results.

Along with generalization of the methods, the other aims and objectives mentioned in the Introduction were automation and dissemination of methods as tools or modules



(a) (b)

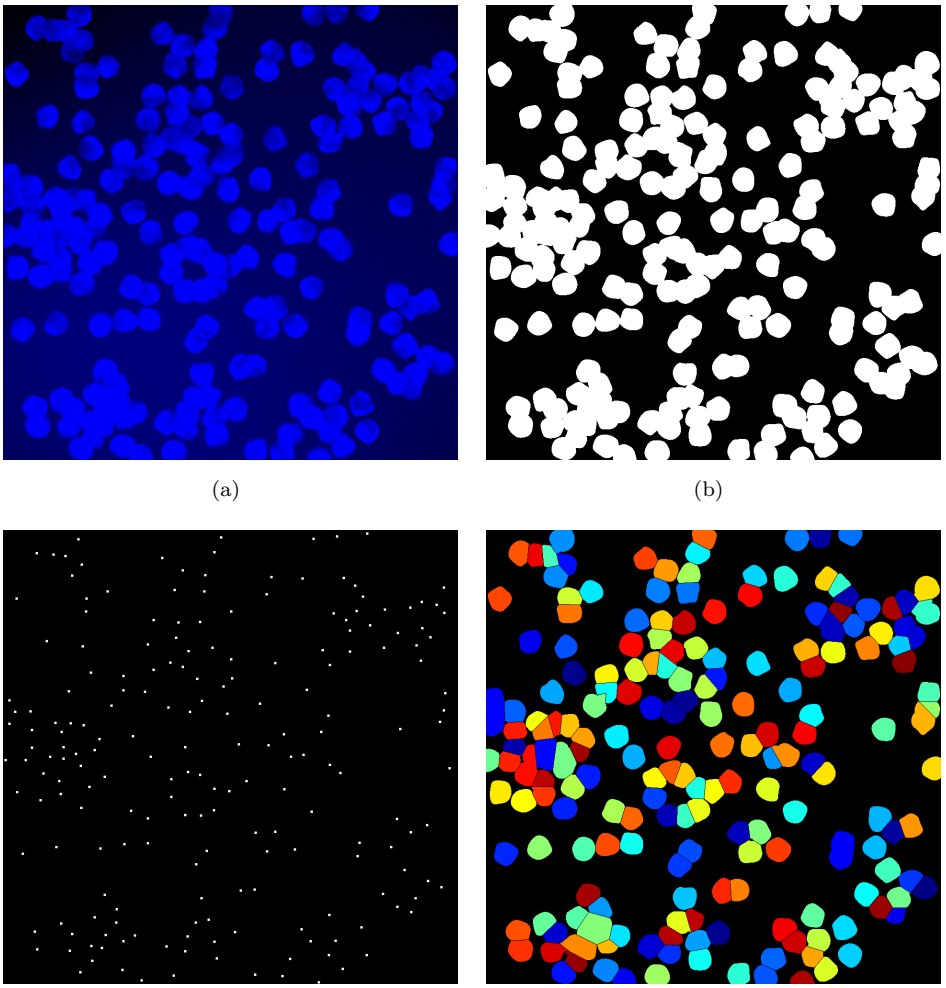
(c) (d)

FIGURE 4.3: Segmentation of budding yeast cell images. (a) A fluorescence microscopy image of budding yeast *S. cerevisiae* cell population from SCMD, (b) its initial segmentation using the method from [66]. (c) and (d) Final segmentation results using clump splitting methods from Sections 2.3.2 and 2.3.3, respectively. The size of the image is 520×696 pixels.

of existing image analysis platforms. The former objective is accomplished quite successfully in the sense that all the three methods do not require any user-defined parameters or threshold values for their operation except that a constraint for the size of the smallest allowed object in the image is used which can be obtained intuitively for even extremely large image sets as is successfully demonstrated in our case study. The latter objective is achieved in a later case study where we made the implementation of the methods of Sections 2.3.2 and 2.3.3 compatible with a widely used cell image analysis platform, i.e., CellProfiler 1.0 [11] such that the methods can be used as modules in other cell image analysis pipelines.

4.2 Whole cell segmentation in high-content screening

High-content screening experiments involving automated fluorescence microscopy imaging captures hundreds of thousands of images in almost no time leaving manual analysis



(a) (b)

(c) (d)

FIGURE 4.4: Segmentation of synthetic cell microscopy images. (a) A synthetic microscopy image of cell population generated from SIMCEP simulation tool [101] and (b) initial segmentation from the tool. (c) Ground truth cell location pointers. (d) Final segmentation result using clump splitting method from Section 2.3.2. The size of the image is 950×950 pixels.

impractical [3]. Besides this, cytometry-based characterization and quantification of cell phenotypes as well as of other cellular or subcellular features require fully automated image analysis methods which are able to analyze the cell microscopy images at single cell level [21, 118]. In order to extract the individual cells from the images, image segmentation is a vital step which needs to be performed in such a way that the final segmentation result contains all the cells separated from the background as well as from each other. These two steps were earlier termed as initial segmentation and clump splitting respectively.

In this case study we apply the developed methods for obtaining whole cell segmentation in high-content screening experiments. As we described earlier, segmentation of cells in high-content screening experiments is typically performed in two levels: cell

nucleus and cell cytoplasm. Segmentation of cell nuclei images resembles more to the problem of segmenting budding yeast cell images because of the regular and often convex shapes of cell nuclei. Therefore, nuclei image segmentation can be performed in a way similar to the one described in the previous subsection. However, due to irregular and rather non-convex shapes of cell cytoplasms, their segmentation is considered to be an application of image segmentation and clump splitting methods described earlier for images containing irregular shaped objects.

In Publication II, we presented a framework for segmentation of cell nuclei images. It begins with image pre-processing using the method presented in Section 2.1. Thereafter, it uses the graph cut image segmentation method of Section 2.2.2 to obtain initial segmentation. A point worth-mentioning is that, later, after development of multi-scale Gaussian representation-based method for initial segmentation discussed in Section 2.2.1, we found that it could also be used for segmenting nuclei images with matching accuracy. Finally, the clump splitting method of Section 2.3.3 is employed in a similar manner discussed in the previous section for splitting clumps of budding yeast cells but with the constraint for the size of smallest allowed object in the image put to 600 pixels for the whole image set. Figure 4.5 shows an image from the test set after the application of the nuclei segmentation procedure outlined above. It also compares the resulting image with results of three state-of-the-art nuclei segmentation methods [11, 21, 58] from literature.

Segmentation of cell cytoplasm images in the context of high-content screening was described in Publication I. A general framework is developed for segmentation and clump splitting of irregular shaped objects. It begins with solving the imaging and cytoplasm images-specific issues by performing image pre-processing as outlined in Section 2.1. Next, the initial segmentation of the pre-processed image is performed with the method described in Section 2.2.1 using this set of values for scale, t = [0, 1, 2, 3, 4, 5, 6]. It should be mentioned that the images we used for evaluation are high-resolution images with low magnification (10x Objective) and this set of values for scale is found appropriate for the other similarly magnified images. Then, the supervised learning-based outline detection method described in Section 2.3.4 is employed along with the nuclei-cytoplasm correspondence-based post-processing described in Section 2.4 which uses the nuclei segmentation described above to separate cell cytoplasms from each other. The ability of modeling technique for yielding sparse models is confirmed since the trained classifier uses only 7 out of 290 pixel-level image features for pixel classification. Figure 4.6 demonstrates the performance of the methodology by showing an image from the test set along with the obtained segmentation result.

Once again, in order to investigate the generalization of the framework we used a challenging set of cell microscopy images with completely different cell characteristics

publicly available at [119] that also facilitates benchmarking. Here, cell nuclei and cytoplasm segmentation is performed in a similar manner described above. Due to relatively

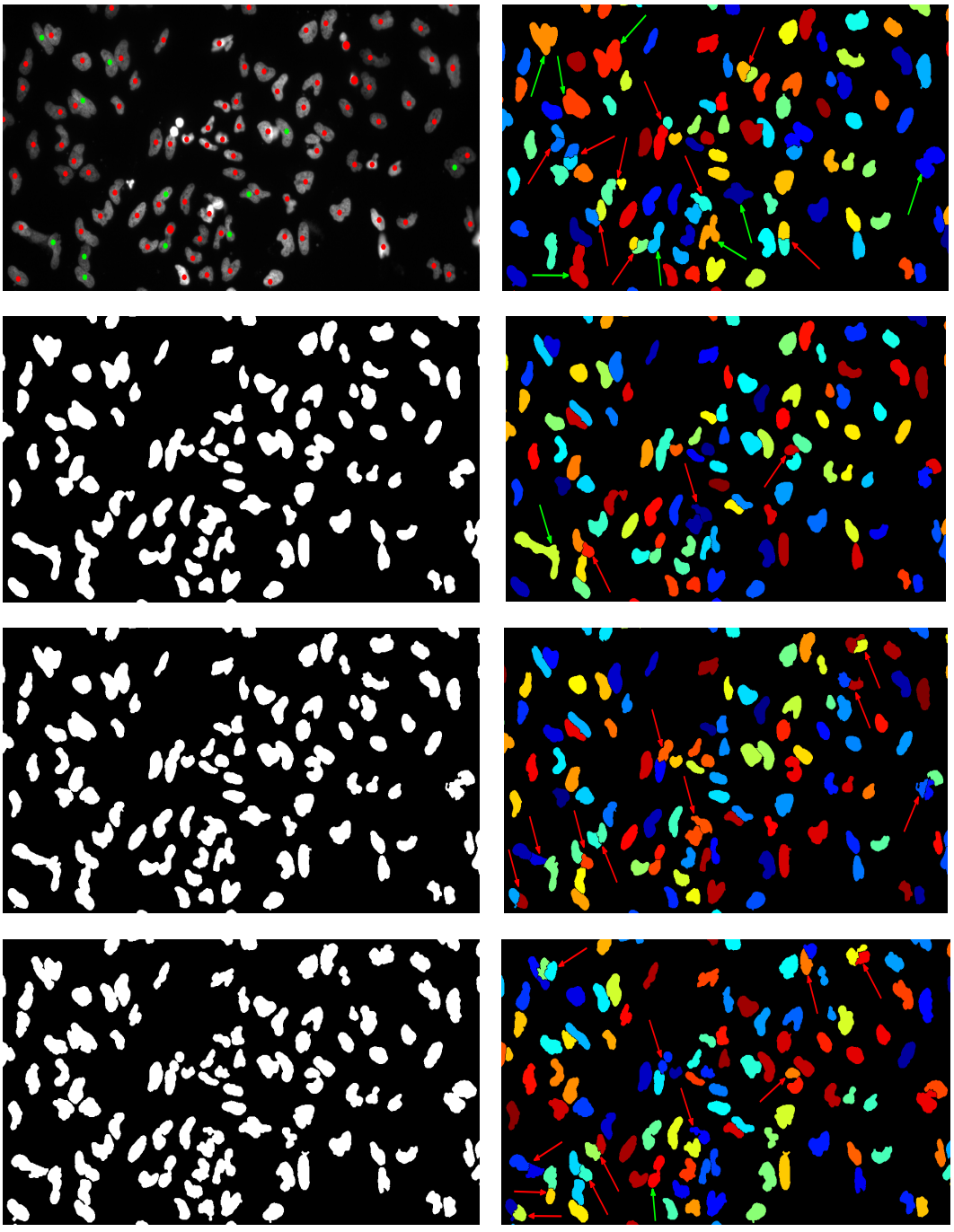
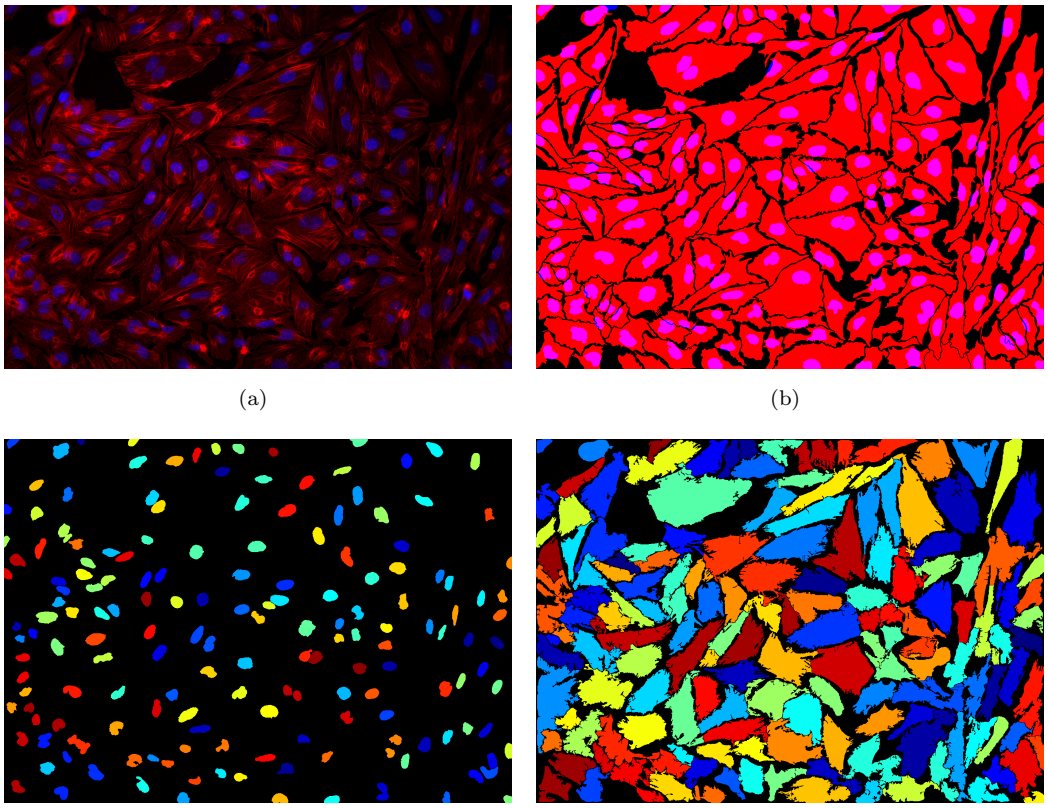


FIGURE 4.5: Qualitative comparison of cell nuclei image segmentation methods. In (top left), a red dot means a separate nucleus and a green dot along with red dot means clump of nuclei. In left Column: (Top to Bottom) Expert labeled image, Initial segmentation results from the method of Section 2.2.2, Level set (LS) and Morphological Gradient (MG) methods. In Right Column: (Top to Bottom) Final segmentation results from CellProfiler and from applying clump splitting method of Section 2.3.3 on the images in the left column. Red and Green arrows indicate over- and under-segmentation respectively.



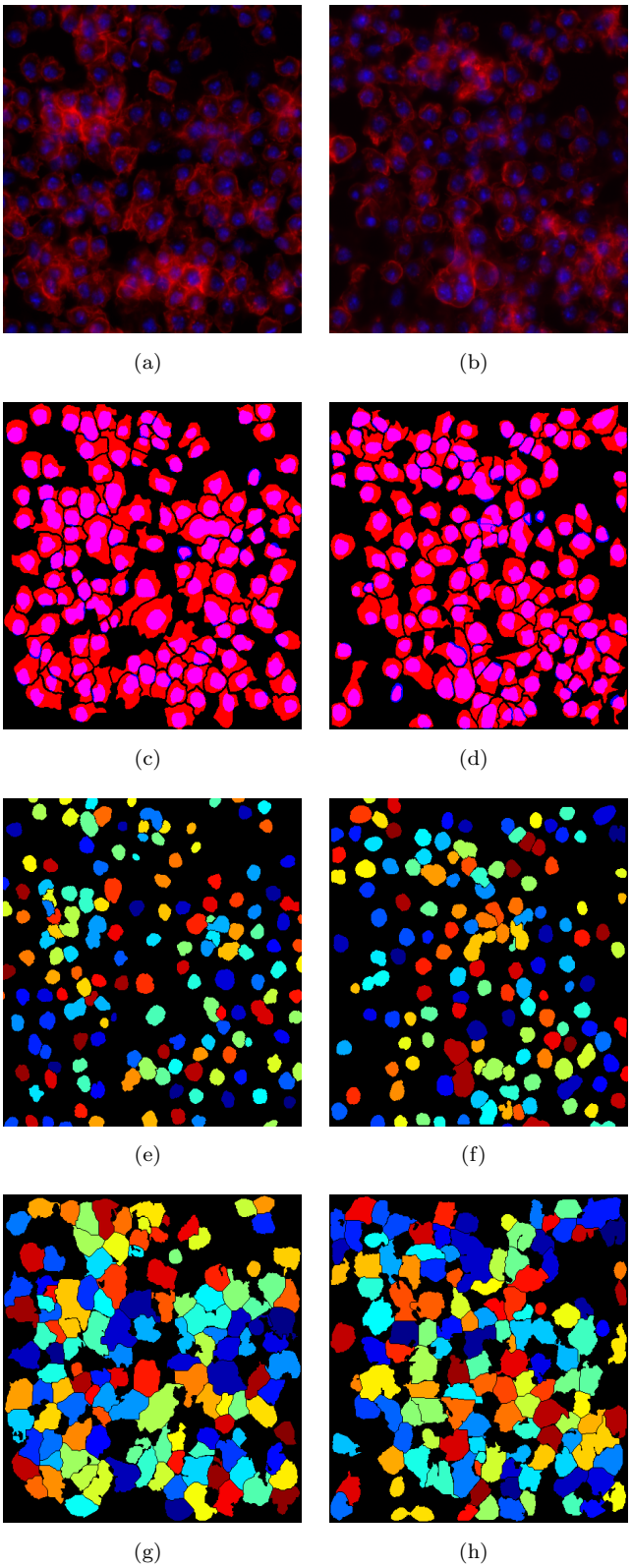
(a) (b)

(c) (d)

FIGURE 4.6: Cell cytoplasm segmentation. (a) A merged cytoplasm (Red)/nuclei (Blue) channel image, (b) benchmark segmentation from biologists, (c) nuclei segmentation and (d) the final segmentation result utilizing methods from Sections 2.2.1 and 2.3.4 along with post-processing. The size of the image is 1040×1392 pixels.

small object sizes, low image resolution and high image magnification not only the size of the smallest allowed object in the image is set to 100 pixels but also the set of values for the scale t used in multi-scale Gaussian representation-based initial segmentation is changed to t = [0, 0.5, 1, 1.5, 2, 2.5, 3]. Here, the classifier trained for outline pixel detection used only 5 out of 290 features for image pixel classification. Two images from this set along with their segmentation results are shown in Figure 4.7 which clearly indicates that the overall framework generalizes quite well. This is also validated by comparing its quantitative measures with those of a recently published method [120] which indicates that the presented framework produces matching or slightly better results.

Finally, it is evident from the above description that the objective of automation and parameter-independence of the methods is fulfilled since the whole framework needs only the selection of a set of scale values to be performed which depends on object sizes, image resolution and magnification and can be done easily. Although the method for cytoplasm segmentation is supervised but the manual input is limited to drawing few outline/non-outline pixels on one or two training images. Moreover, a supplementary



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 4.7: Cell cytoplasm segmentation. (a)-(b) Two merged cytoplasm (Red)/nuclei (Blue) channel images, (c)-(d) benchmark images, (e)-(f) nuclei segmentation and (g)-(h) the final segmentation result utilizing methods from Sections 2.2.1 and 2.3.4 along with post-processing. The size of the images is 450×450 pixels.

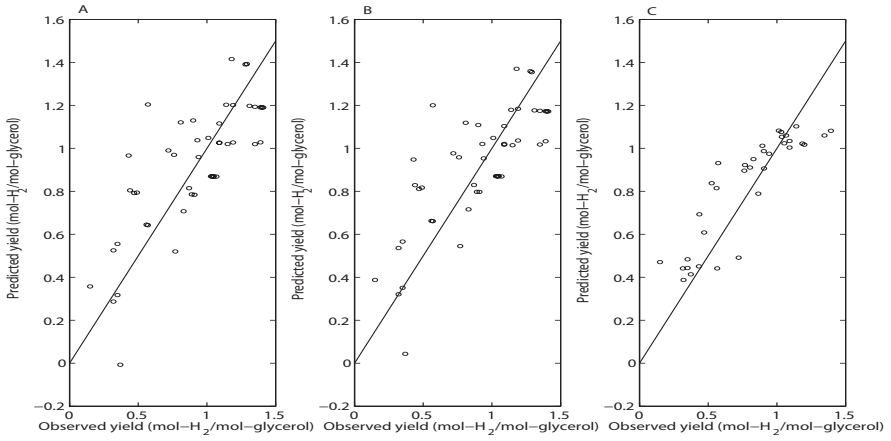
website for Publication I presents the codes and all the other relevant details about implementation helping the dissemination of methods to the community for usage in their analysis pipelines as well as for further extension and comparisons.

4.3 Bioprocess data mining and scale-up modeling

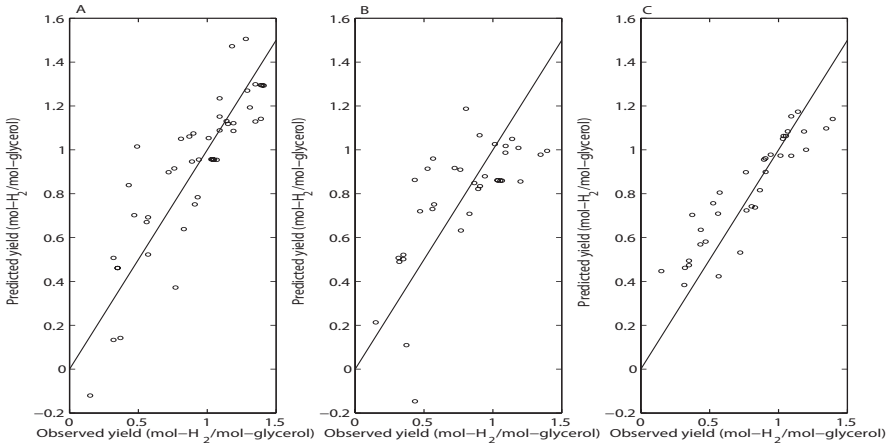
In Publication V, we used the data from bioconversion of crude glycerol to hydrogen, a study related to culture media optimization (unpublished data, Rahul Mangayil *et al.*), as a case study for evaluating the modeling and data mining methods presented in Section 3.1 for yield prediction. The data consist of 35 samples with five operational parameters involved in the process design. Along with those five parameters we also investigated the usage of their first and the second order polynomials in the data model in order to model the non-linearity in the data, if it exists at all. In this test case, leave-one-out cross-validation was used to estimate the prediction accuracy, given in terms of the correlation between the actual and predicted output value, for optimal model selection. For the method based on regularized regression, the correlation is a bit higher in the case of using non-linearly transformed data along with the actual data as opposed to using only actual data which indicates non-linearity inherent in the data. The method based on random forests, however, is able to give even better correlation measures because of its capability of handling non-linearity of the data. However, it is difficult or rather inappropriate to judge the methods based on the small difference in prediction accuracy for this dataset, especially, knowing that the embedded feature selection property of regularized regression makes it worthy in more complicated and high-dimensional data analysis. Figure 4.8 shows the results of the three different modeling approaches with and without using non-linearly transformed variables in modeling.

In order to evaluate the scale-up methodology we used a case study in Publication VI that contained 117 samples from different scale experiments producing a cytotoxic compound called anthracycline. Experiments were performed in flasks (81 samples), 2L fermenters (24 samples), and 30L fermenters (12 samples) with a total of around 40 variables, both numerical and categorical, involved in the process. Dummy coding increased the number of variables to be used in process modeling to more than 70. First, data modeling is performed using regularized regression-based method for predicting the product yield which produced a model with only 30 non-zero coefficients out of 73 model coefficients. Thus automatic selection of important variables is performed without any *a priori* biological knowledge of the process. Figure 4.9 compares the experimentally observed and the predicted values of the product yield. It is clear that despite all the technical challenges the methodology is able to produce a quite accurate model.

Next, the scale-up is investigated by using flask and 2L samples as two alternative



(a)



(b)

FIGURE 4.8: Modeling approaches to model production of hydrogen. The results of the three different modeling approaches: A. linear regression, B. regularized regression and C. Random forests (a) without and (b) with the usage of non-linearly transformed variables in modeling

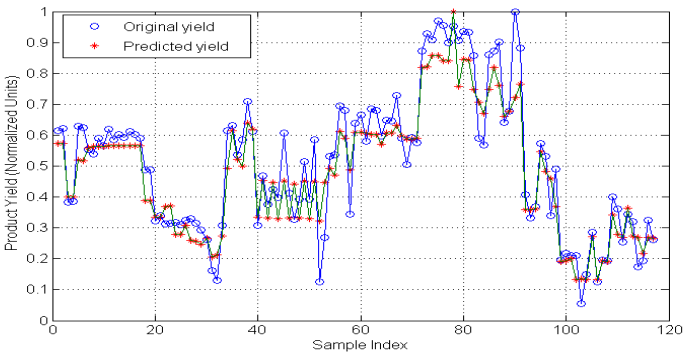
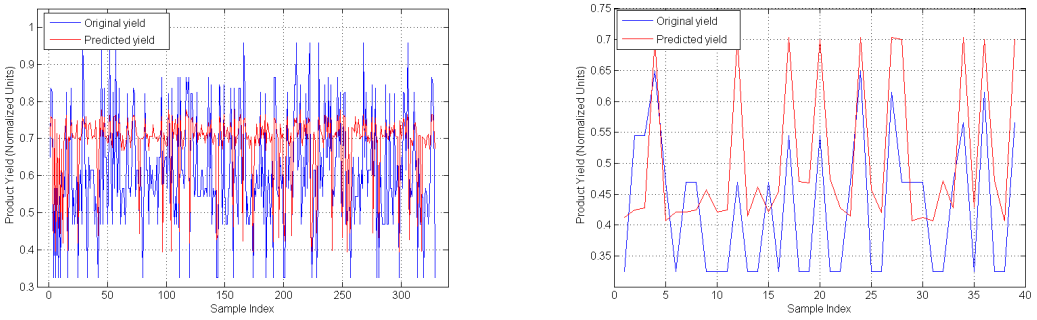


FIGURE 4.9: Result of product yield (normalized to the range (0, 1]) prediction model for the samples from flask experiments compared to the experimentally observed product yields.

small scale data and 30L samples as large scale data. The tolerance range for product yield correspondence, i.e., ϵ is set to 0.2. The data rearrangement resulted in 330 and 39 measurement sample-pairs in flask and 2L cases, respectively. Both the small scale



(a) (b)

FIGURE 4.10: The predicted product yield (normalized to the range (0, 1]) after scale-up to 30L experiment from (a) flask experiment and (b) 2L experiment, compared to the experimentally observed product yield of 30L experiment samples.

data samples when used with the scale-up methodology described in Section 3.2 yielded corresponding predicted large scale samples. Then the already produced product yield model was used to predict the values of the product yield for the predicted large scale samples. Comparison of the experimental and the predicted product yield is made in Figure 4.10 where it is clear that, in both the scale-up alternatives, for most of the samples the difference between the product yields is within the tolerance range of 0.2 units. This almost matching accuracy also suggests that 2L fermentation experiments can be skipped in future to improve the cost efficiency of the bioprocess development.

Chapter 5

Discussion

This thesis primarily focuses on facilitating high-throughput imaging-based experiments with automated quantitative image analysis. It presents automated and to some extent non-parametric and generalized methods answering the underlying challenges in image analysis for a wide range of applications, with the emphasis on cytometry in biomedical microscopy applications. The initial segmentation methods based on multi-scale Gaussian and graph cut and the clump splitting method based on supervised learning using large generic set of pixel-level features find applications in many general studies performing object-level analysis involving both regular and irregular shaped objects. Whereas the concavity point analysis-based methods for clump splitting though limited to images containing convex shaped objects find usage in cell microscopy as well as in many industrial applications. From the biomedical microscopy application point of view, the methods help in more accurate detection of individual cells leading to more exact feature extraction, measurements, classification etc. facilitating single-cell analysis, e.g., for quantification of cell phenotypes.

With the aim of automation, dissemination of methods for usage in routine biological analysis, generalization of the methods to other applications and fulfillment of high-throughput analysis in mind, the approach more suitable for handling complex image analysis tasks is to separate the different analysis steps into modules of an analysis pipeline. For cell image analysis, CellProfiler [11, 121] is a widely used platform with the provision of creating customized analysis pipelines using the built-in or self-created modules. Along with the creation of new modules, there is the possibility of expansion and modification in the existing ones to improve their performance and/or to design an application-specific module. Moreover, this also provides a way for further evaluation of the developed image analysis methods based on the subsequent biological results obtained from the full pipelines. With CellProfiler being used frequently in the biologist community it provides a solid platform to integrate the developed methods e.g. [122],

once validated, in the community for usage in routine analysis thus fulfilling the desire for their development. This study proceeds along similar lines where all the methods, except the supervised learning-based clump splitting method, have been implemented such that they can be used as modules of CellProfiler 1.0 and are freely available to be used directly in analysis pipelines.

Along with the fluorescence microscopy-based high-throughput, high-content screening, the other emerging technology is lab-on-a-chip microfluidics[123, 124] performing high-throughput live cell and subcellular level imaging experiments [17, 19, 125–127]. The on-going advancements in both these fields and in biomedical research, where poses some new image analysis challenges, also provides the image analysts and developers with the opportunity to improve existing as well as to develop new computational methods for more faster and accurate quantitative analysis. The methods from this study are supposed to work appropriately for live single cell imaging in microfluidics lab-on-a-chip platform, yet it needs to be investigated properly in future.

Although, a typical cell image analysis pipeline involves feature extraction, measurement and classification after cell detection but most of these tasks are application-specific or even general frameworks conform better to these tasks. A more complementary task, however, for cell detection, especially in the live cell imaging platform of microfluidics, is tracking of single cells in time-series images over time. Despite the abundance of object tracking methods that have been used in cell and subcellular object tracking [16–18, 49, 128–130], the requirement of doing it precisely for high-throughput live cell imaging is extremely challenging where besides dealing with imaging aberrations the other challenges are cells dividing and moving inside and outside of the field of view. Therefore, implementation of object tracking methods answering these challenges is a more appropriate continuation of this study in future.

A secondary focus of this thesis is to cope with the challenges in data mining and analysis of industrial biological processes to effectively scale them up from small scale laboratory experiments to large industrial scales. The study takes up the data mining and analysis problems as statistical modeling tasks which meant the developed methods can be generalized to analyze other similar processes more so because no *a priori* information about the process is used for creating the models. Moreover, the automatic selection of the most important process operational parameters affecting the product yield helps in determining advantageous control direction. In certain applications, studying the interaction between operational parameters and incorporating them into modeling gives more insight into process control. One of the challenges in these studies is high-dimensional data which hinders proper visualizations of the variable space to provide overview of the models. Also different experimental setting at different scales causes missing values which can be problematic in defining scale-up models and needs to be investigated with different experimental data in future. One of the key steps forward

is the utilization of imaging-based in-line sensors for monitoring of important process variables so that cytometry-based cell characterization can be used in modeling and controlling of the bioprocess. Finally, generation of more case studies involving several process types producing different organisms and products would lead to a more detailed characterization of the methodology and help in improving it.

Errata for the publications

- In Publication I, subsection "Design of classifier incorporating feature selection", the classes "outline" and "non-outline" given as o_i and n_i respectively should not have the subscript i.
- In Publication V, the second paragraph of the subsection "Multiple linear regression" starting with "In spite of being linear" should instead be read as "Because of being linear".

Bibliography

[1] R. Pepperkok and J. Ellenberg. High-throughput fluorescence microscopy for systems biology. *Nature Reviews Molecular Cell Biology*, 7(9):690–696, 2006.

[2] A. Allalou, F. M. van de Rijke, R. J. Tafrechi, A. K. Raap, and C. Wählby. Image based measurements of single cell mtDNA mutation load. *Springer LNCS*, 4522: 631–640, 2007.

[3] R. Wollman and N. Stuurman. High throughput microscopy: from raw images to discoveries. *Journal of Cell Science*, 120(21):3715–3722, 2007.

[4] S. A Haney, P. LaPan, J. Pan, and J. Zhang. High-content screening moves to the front of the line. *Drug Discovery Today*, 11(19-20):889–894, 2006.

[5] Attila Tárnok. A focus on high-content cytometry. *Cytometry Part A*, 73(5): 381–383, 2008.

[6] X. Zhou, X. Zhou, and S. Wong. High content cellular imaging for drug development. *IEEE Signal Processing Magazine*, 23(2):170–174, 2006.

[7] J. Wang, X. Zhou, P. L. Bradley, S.-F. Chang, N. Perrimon, and S. T. C. Wong. Cellular phenotype recognition for high-content RNA interference genome-wide screening. *Journal of Biomolecular Screening*, 13(1):29–39, 2008.

[8] J. Z. E. Perlman, M. D. Slack, Y. Feng, T. J. Mitchison, L. F. Wu, and S. J. Altschuler. Multidimensional drug profiling by automated microscopy. *Science*, 306(5699):1194–1198, 2004.

[9] O. M Dzyubachyk, W. J Niessen, and E. Meijering. Advanced level-set based multiple-cell segmentation and tracking in time-lapse fluorescence microscopy images. In *Proc. IEEE International Symposium on Biomedical Imaging*, pages 185–188, 2008.

[10] X. Zhou and S. Wong. Informatics challenges of high-throughput microscopy. *IEEE Signal Processing Magazine*, 23:63–72, 2006.

[11] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10):R100, 2006.

[12] C. Chen, H. Li, X. Zhou, and S. T. C. Wong. Constraint factor graph cut based active contour method for automated cellular image segmentation in RNAi screening. *Microscopy*, 230(2):177–191, 2008.

[13] P. Ruusuvuori. *Methods for image analysis, object classification and validation in biomedical microscopy applications.* PhD thesis, Tampere University of Technology, Finland, 2009.

[14] A. Niemistö. *Quantitative image analysis methods for applications in biomedical microscopy.* PhD thesis, Tampere University of Technology, Finland, 2006.

[15] C. Wählby, I.-M. Sintorn, P. Erlandsson, G. Borgefors, and E. Bengtsson. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. *Microscopy*, 215(1):67–76, 2004.

[16] I. Smal, D. Draegestein, N. Galjart, W. Niessen, and E. Meijering. Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: application to microtubule growth analysis. *IEEE Transactions on Medical Imaging*, 27 (6):789–804, 2008.

[17] D. Falconnet, A. Niemistö, R. J. Taylor, M. Ricicova, T. Galitski, I. Shmulevich, and C. L. Hansen. High-throughput tracking of single yeast cells in a microfluidic imaging matrix. *Lab on a Chip*, 11(3):466–473, 2011.

[18] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, and G. Danuser. Robust single particle tracking in live cell time-lapse sequences. *Nature Methods*, 5(5):695–702, 2008.

[19] R. J. Taylor, D. Falconnet, A. Niemistö, S. A. Ramsey, S. Prinz, I. Shmulevich, T. Galitski, and C. L. Hansen. Dynamic analysis of MAPK signaling using a high-throughput microfluidic single-cell imaging platform. In *Proc. National Academy of Sciences of USA*, pages 3758–3763, 2009.

[20] F. Li, X. Zhou, J. Zhu, J. Ma, and S. T. C. Wong. Workflow and methods of high-content time-lapse analysis for quantifying intracellular calcium signals. *Neuroinformatics*, 6(2):97–108, 2008.

[21] P. Matula, A. Kumar, I. Wörz, H. Erfle, R. Bartenschlager, R. Eils, and K. Rohr. Single-cell-based image analysis of high-throughput cell array screens for quantification of viral infection. *Cytometry*, 75(4):309–318, 2009.

[22] X. Zhou, F. Li, J. Yan, and S. T. Wong. A novel cell segmentation method and cell phase identification using markov model. *IEEE Transactions on Information Technology in Biomedicine*, 13(2):152–157, 2009.

[23] D. Padfield, J. Rittscher, N. Thomas, and B. Roysam. Spatio-temporal cell cycle phase analysis using level sets and fast marching methods. *Medical Image Analysis*, 13(1):143–155, 2009.

[24] W. Wallace, L. H. Schaefer, and J. R. Swedlow. A workingperson's guide to deconvolution in light microscopy. *BioTechniques*, 31(5):1076–82, 2001.

[25] O. Daněk, P. Matula, C. Ortiz de Solórzano, A. Muñoz-Barrutia, M. Maška, and M. Kozubek. Segmentation of touching cell nuclei using a two-stage graph cut model. *Springer LNCS*, 5575:410–419, 2009.

[26] C. Wählby, J. Lindblad, M. Vondrus, E. Bengtsson, and L. Björkesten. Algorithms for cytoplasm segmentation of fluorescence labelled cells. *Analytical Cellular Pathology*, 24(2-3):101–111, 2002.

[27] C. D. Ruberto, A. Dempster, S. Khan, and B. Jarra. Segmentation of blood images using morphological operators. In *Proc. IEEE International Conference on Pattern Recognition*, pages 397–400, 2000.

[28] M. Farhan. Automated clump splitting for biological cell segmentation in microscopy using image analysis. Master's thesis, Tampere University of Technology, Finland, 2010.

[29] J. Selinummi. *On algorithms for two and three dimensional high throughput light microscopy*. PhD thesis, Tampere University of Technology, Finland, 2008.

[30] G. Fernandez, M. Kunt, and J.-P. Zrÿd. A new plant cell image segmentation algorithm. In *Proc. 8th International Conference on Image Analysis and Processing*, pages 229–234, 1995.

[31] W. Wang and H. Song. Cell cluster image segmentation on form analysis. In *Proc. 3rd International Conference on Natural Computation*, pages 833–836, 2007.

[32] J. Liang. Intelligent splitting in the chromosome domain. *Pattern Recognition*, 22 (5):519–532, 1989.

[33] Q. Wen, H. Chang, and B. Parvin. A Delaunay triangulation approach for segmenting clumps of nuclei. In *Proc. IEEE International Symposium on Biomedical Imaging*, pages 9–12, 2009.

[34] J. Selinummi, P. Ruusuvuori, I. Podolsky, A. Ozinsky, E. Gold, O. Yli-Harja, A. Aderem, and I. Shmulevich. Bright field microscopy as an alternative to whole cell fluorescence in automated analysis of macrophage images. *PLoS One*, 4(10): e7497, 2009.

[35] A. Niemistö, T. Aho, H. Thesleff, M. Tiainen, K. Marjanen, M.-L. Linne, and O. Yli-harja. Estimation of population effects in synchronized budding yeast experiments. In *Proc. SPIE 2003. Image Processing: Algorithms and Systems II*, pages 448–459, 2003.

[36] A. Niemistö, J. Selinummi, R. Saleem, I. Shmulevich, J. Aitchison, and O. Yli-Harja. Extraction of the number of peroxisomes in yeast cells by automated image analysis. In *Proc. IEEE International Conference on Engineering in Medicine and Biology Society*, pages 448–459, 2006.

[37] N. Malpica, C. Ortiz de Solórzano, J. J. Vaquero, A. Santos, S. J. Lockett, I. Vallcorba, J. M. Garcia-Sagredo, and F. D. Pozo. Applying watershed algorithms to the segmentation of clustered nuclei. *Cytometry*, 28:289–297, 1997.

[38] P. Yan, X. Zhou, M. Shah, and S. T. C. Wong. Automatic segmentation of high-throughput RNAi fluorescent cellular images. *IEEE Transactions on Information Technology in Biomedicine*, 12(1):109–117, 2008.

[39] J. A. Rocha-Valadez, M. Estrada, E. Galindo, and L. Serrano-Carreon. From shake flasks to stirred fermenters: Scale-up of an extractive fermentation process for 6-pentyl-a-pyrone production by trichoderma harzianum using volumetric power input. *Process Biochemistry*, 41:1347–1352, 2006.

[40] W. Katzer, M. Blackburn, K. Charman, S. Martin, J. Penn, and S. Wrigley. Scale-up of filamentous organisms from tubes and shake-flasks into stirred vessels. *Biochemical Engineering*, 7:127–134, 2001.

[41] CDER: Process validation: General principles and practices. URL `http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/ucm070336.pdf`.

[42] H. B. Junker. Scale-up methodologies for Eschericia coli and yeast fermentation processes. *Bioscience and Bioengineering*, 97:347–364, 2004.

[43] F. R. Schmidt. Optimization and scale up of industrial fermentation processes. *Applied Microbiology and Biotechnology*, 68:425–435, 2005.

[44] R. C. Gonzalez and R. E. Woods. *Digital image processing.* 3rd ed. Prentice Hall, 2008.

[45] J. C. Russ. *The image processing handbook*. 4th ed. CRC Press, 2002.

[46] W. X. Wang. Binary image segmentation of aggregates based on polygonal approximation and classification of concavities. *Pattern Recognition*, 31(10):1503–1524, 1998.

[47] M. Leskó, Z. Kato, A. Nagi, I. Gombos, Z. Török, L. Vigh, and L. Vigh. Live cell segmentation in fluorescence microscopy via graph cut. In *Proc. IEEE International Conference on Pattern Recognition*, pages 1485–1488, 2010.

[48] C. Sommer, C. Straehle, U. Kothe, and F. A. Hamprecht. Ilastik: interactive learning and segmentation toolkit. In *Proc. IEEE International Symposium on Biomedical Imaging*, pages 230–233, 2011.

[49] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J.-C. Olivo-Marin. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. *IEEE Transactions on Medical Imaging*, 21(10):1212–1221, 2002.

[50] J. Kolega. The movement of cell clusters in vitro: morphology and directionality. *Journal of Cell Science*, 49:15–32, 1981.

[51] O. Schmitt and M. Hasse. Radial symmetries based decomposition of cell clusters in binary and gray level images. *Pattern Recognition*, 41(6):1905–1923, 2008.

[52] A. Garrido and N. P. de la Blanca. Applying deformable templates for cell image segmentation. *Pattern Recognition*, 33(5):821–832, 2000.

[53] E. Bengtsson, C. Wählby, and J. Lindblad. Robust cell image segmentation methods. *Pattern Recognition and Image Analysis*, 14:157–167, 2004.

[54] A. Bleau and L. Leon. Watershed-based segmentation and region merging. *Computer Vision and Image Understanding*, 77(3):317–370, 2000.

[55] C. Garbay, J. M. Chassery, and G. Brugal. An iterative region-growing process for cell image segmentation based on local color similarity and global shape criteria. *Analytical and Quantitative Cytology and Histology*, 8:25–34, 1986.

[56] D. L. Pham, C. Xu, and J. L. Prince. Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337, 2000.

[57] T. Brox and J. Weickert. Level set based image segmentation with multiple regions. *Pattern Recognition, Springer LNCS*, 3175:415–423, 2004.

[58] C. Li, C. Xu, C. Gui, and M. D. Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing*, 19(12):3243–3254, 2010.

[59] O. Bernard, D. Friboulet, P. Thevenaz, and M. Unser. Variational B-spline level-set method for fast image segmentation. In *Proc. IEEE International Symposium on Biomedical Imaging*, pages 177–180, 2008.

[60] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam. Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*, 57(4):841–852, 2010.

[61] J. Lindblad, C. Wählby, E. Bengtsson, and A. Zaltsman. Image analysis for automatic segmentation of cytoplasms and classification of Rac1 activation. *Cytometry, Part A*, 57(1):22–33, 2004.

[62] K. Zuiderveld. *Contrast limited adaptive histogram equalization.* Graphic Gems IV San Diego: Academic Press Professional, 1994.

[63] T. R. Jones, A. E. Carpenter, and P. Golland. Voronoi-based segmentation of cells on image manifolds. In *ICCV Workshop on Computer Vision for Biomedical Image Applications*, pages 535–543, 2005.

[64] G. Xiong, X. Zhou, L. Ji, P. Bradley, N. Perrimon, and S. Wong. Segmentation of drosophila RNAi fluorescence images using level sets. In *Proc. IEEE International Conference on Image Processing*, pages 73–76, 2006.

[65] P. Ruusuvuori, J. Seppälä, T. Erkkilä, A. Lehmussola, J. A. Puhakka, and O. Yli-harja. Efficient automated method for image-based classification of microbial cells. In *Proc. IEEE International Conference on Pattern Recognition*, pages 1–4, 2008.

[66] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[67] T. Lindeberg. Scale-space theory: a basic tool for analysing structures at different scales. *Journal of Applied Statistics, Supplement Advances in Applied Statistics: Statistics and Images: 2*, 21(2):225–270, 1994.

[68] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[69] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[70] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[71] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proc. IEEE International Conference on Computer Vision*, pages 26–33, 2003.

[72] S. Eddins. Cell segmentation, 2013. URL `http://blogs.mathworks.com/steve/2006/06/02/cell-segmentation`.

[73] G. Guo, X. Ping, D. Hu, and J. Yang. An efficient segmentation algorithm based on mathematical morphology and improved watershed. *Intelligent Computing in Signal Processing and Pattern Recognition, Lecture Notes in Control and Information Sciences*, 345:689–695, 2006.

[74] N. Lu and X. Ke. A segmentation method based on gray-scale morphological filter and watershed algorithm for touching objects image. In *Proc. 4th International Conference on Fuzzy Systems and Knowledge Delivery*, pages 474–478, 2007.

[75] N. Sweeney and B. V. Sweeney. Efficient segmentation of cellular images using gradient-based methods and simple morphological filters. In *Proc. IEEE International Conference on Engineering in Medicine and Biology Society*, pages 880–882, 1997.

[76] X. Bai, C. Sun, and F. Zhou. Splitting touching cells based on concave points and ellipse fitting. *Pattern Recognition*, 42(11):2434–2446, 2009.

[77] G. Cong and B. Parvin. Model-based segmentation of nuclei. *Pattern Recognition*, 33(8):1383–1393, 2000.

[78] S. Kothari, Q. Chaudry, and M. D. Wang. Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques. In *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 795–798, 2009.

[79] L. Shen, X. Song, M. Iguchi, and F. Yamamoto. A method for recognizing particles in overlapped particle images. *Pattern Recognition Letters*, 21:21–30, 2000.

[80] C. Ortiz de Solórzano, R. Malladi, S. A. Lelievre, and S. J. Lockett. Segmentation of nuclei and cells using membrane related protein markers. *Microscopy*, 201: 404–415, 2001.

[81] G. Zhang, D. S. Jayas, and N. D. G. White. Separation of touching grain kernels in an image by ellipse fitting algorithm. *Biosystems Engineering*, 92(2):135–142, 2005.

[82] S. Kumar, S. H. Ong, S. Ranganath, T. C. Ong, and F. T. Chew. A rule-based approach for robust clump splitting. *Pattern Recognition*, 39(6):1088–1098, 2006.

[83] S. Raman, B. Parvin, C. Maxwell, and M. H. Barcellos-Hoff. Geometric approach to segmentation and protein localization in cell cultured assays. *Microscopy*, 225 (1):22–30, 2007.

[84] Q. Zhong, P. Zhou, Q. Yao, and K. Mao. A novel segmentation algorithm for clustered slender-particles. *Computer and Electronics in Agriculture*, 69(2):118–127, 2009.

[85] M. Brejl and M. Sonka. Edge based image segmentation: machine learning from examples. In *Proc. IEEE International Conference on Neural Networks. IEEE World Congress on Computational Intelligence*, pages 814–819, 1998.

[86] M. Prasad, A. Zisserman, A. Fitzgibbon, M. P. Kumar, and P. H. S. Torr. Learning class-specific edges for object detection and segmentation. In *Proc. Indian Conference on Computer Vision, Graphics and Image Processing*, pages 94–105, 2006.

[87] M. Keuper, R. Bensch, K. Voigt, A. Dovzhenko, K. Palme, H. Burkhardt, and O. Ronneberger. Semi-supervised learning of edge filters for volumetric image segmentation. In *DAGM-Symposium*, pages 462–471, 2010.

[88] M. Brejl and M. Sonka. Object localization and border detection criteria design in edge-based image segmentation: automated learning from examples. *IEEE Transactions on Medical Imaging*, 19(10):973–985, 2000.

[89] R. Duda, P. Hart, and D. Stork. *Pattern classification.* John Wiley and Sons, 2001.

[90] P. Ruusuvuori, T. Manninen, and H. Huttunen. Image segmentation using sparse logistic regression with spatial prior. In *Proc. IEEE European Signal Processing Conference*, pages 2253–2257, 2012.

[91] E. Tuv, A. Borisov, G. Runger, and K. Torkkola. Feature selection with ensembles, artificial variables, and redundancy elimination. *Machine Learning Research*, 10: 1341–1366, 2009.

[92] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Royal Statistical Society Series B Methodological*, 58:267–288, 1996.

[93] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: Data mining, inference, and prediction.* Springer Series in Statistics, Springer, 2009.

[94] P. K. Andersen and L. T. Skovgaard. *Multiple regression, the linear predictor. In regression with linear prediction.* New York, NY: Springer, 2010.

[95] J. H. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[96] H. Huttunen, J.-P. Kauppi, and J. Tohka. Regularized logistic regression for mind reading with parallel validation. In *Winning submission to Mind reading from MEG, PASCAL Challenge in ICANN*, pages 20–24, 2011.

[97] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[98] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, pages 14–19, 1990.

[99] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[100] D. Webb, M. A. Hamilton, G. J. Harkin, S. Lawrence, A. K. Camper, and Z. Lewandowski. Assessing technician effects when extracting quantities from microscope images. *Journal of Microbiological Methods*, 53(1):97–106, 2003.

[101] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja. Computational framework for simulating fluorescence microscope images with cell populations. *IEEE Transactions on Medical Imaging*, 26:1010–1016, 2007.

[102] A. Lehmussola, P. Ruusuvuori, and O. Yli-Harja. Evaluating the performance of microarray segmentation algorithms. *Bioinformatics*, 22(23):2910–2917, 2006.

[103] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8): 861–874, 2006.

[104] M. P. C. Marques, J. M. S. Cabral, and P. Fernandes. Bioprocess scale-up: quest for the parameters to be used as criterion to move from microreactors to lab-scale. *Chemical Technology and Biotechnology*, 85:1184–1198, 2010.

[105] J. M. Seletzky, U. Noak, J. Fricke, E. Welk, W. Eberhard, and C. Knocke et al. Scale-up from shake flasks to fermenters in batch and continuous mode with corynebacterium glutamicum on lactic acid based on oxygen transfer and pH. *Biotechnology and Bioengineering*, 98:800–811, 2007.

[106] F. Garcia-Ochoa and E. Gomez. Bioreactor scale-up and oxygen transfer rate in microbial processes: An overview. *Biotechnology Advances*, 27:153–176, 2009.

[107] N. Mehmood, E. Olmos, P. Marchal, J.-L. Goergen, and S. Delaunay. Relation between pristinamycins production by streptomyces pristinaespiralis, power dissipation and volumetric gas-liquid mass transfer coefficient, k_La. *Process Biochemistry*, 45:1779–1786, 2010.

[108] Y.-L. Hsu and W.-T. Wu. A novel approach for scaling-up a fermentation system. *Biochemical Engineering*, 11:123–130, 2002.

[109] S. Ogawa, T. Kamijima, Y. Miyamotao, M. Miyajima, H. Sato, and K. Takayama et al. A new attempt to solve the scale-up problem for granulation using response surface methodology. *Pharmaceutical Sciences*, 83:439–443, 1994.

[110] S. Saran, J. Isar, and R. K. Saxena. Statistical optimization of conditions for protease production from Bacillus sp. and its scale-up in a bioreactor. *Applied Biochemistry and Biotechnology*, 141:229–240, 2007.

[111] D. W. Stockburger. *Multivariate statistics: concepts, models, and applications.* Missouri State University, 2001.

[112] J.-P. Kauppi, H. Huttunen, H. Korkala, I. P. Jääskeläinen, M. Sams, and J. Tohka. Face prediction from fMRI data during movie stimulus: Strategies for feature selection. In *Proc. International Conference on Artificial Neural Networks ICANN*, pages 189–196, 2011.

[113] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[114] L. Breiman. *Classification and regression trees.* CRC press, 1993.

[115] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[116] RF-ACE : multivariate machine learning with heterogeneous data. URL `http://code.google.com/p/rf-ace/`.

[117] T. L. Saito, M. Ohtani, H. Sawai, F. Sano, A. Saka, D. Watanabe, M. Yukawa, Y. Ohya, and S. Morishita. SCMD: Saccharomyces cerevisiae morphological database. *Nucleic Acids Research*, 32:D319–D322, 2004.

[118] J. Selinummi, J. R. Sarkanen, A. Niemistö, M. L. Linne, T. Ylikomi, O. Yli-Harja, and T. O. Jalonen. Quantification of vesicles in differentiating human SH-SY5Y neuroblastoma cells by automated image analysis. *Neuroscience Letters*, 396(2): 102–107, 2006.

[119] Broad bioimage benchmark collection website, 2013. URL `http://www.broadinstitute.org/bbbc/BBBC007/`.

[120] P. Quelhas, M. Marcuzzo, A. M. Mendonça, and A. Campilho. Cell nuclei and cytoplasm joint segmentation using the sliding band filter. *IEEE Transactions on Medical Imaging*, 29(8):1463–1473, 2010.

[121] T. R. Jones, I. H. Kang, D. B. Wheeler, R. A. Lindquist, A. Papallo, D. M. Sabatini, P. Golland, and A. E. Carpenter. Cellprofiler analyst: data exploration and analysis software for complex image-based screens. *BMC Bioinformatics*, 9 (1):482, 2008.

[122] P. Rämö, R. Sacher, B. Snijder, B. Begemann, and L. Pelkmans. CellClassifier: supervised learning of cellular phenotypes. *Bioinformatics*, 25(22):3028–30, 2009.

[123] G. M. Whitesides. The origins and the future of microfluidics. *Nature*, 442:368–373, 2006.

[124] D. N. Breslauer, P. J. Lee, and L. P. Lee. Microfluidics-based systems biology. *Molecular Biosystems*, 2:97–112, 2006.

[125] P. J. Lee, N. C. Helman, W. A. Lim, and P. J. Hung. A microfluidic system for dynamic yeast cell imaging. *BioTechniques*, 44(1):91–95, 2008.

[126] M. C. Park, J. Y. Hur, H. S. Cho, S.-H. Park, and K. Y. Suh. High-throughput single-cell quantification using simple microwell-based cell docking and programmable time-course live-cell imaging. *Lab Chip*, 11(1):79–86, 2011.

[127] D. R. Albrecht, G. H. Underhill, J. Resnikoff, A. Mendelson, S. N. Bhatia, and J. V. Shah. Microfluidics-integrated time-lapse imaging for analysis of cellular dynamics. *Integrative Biology*, 2:278–287, 2010.

[128] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:51–65, 2005.

[129] C. Chen, X. Zhou, and S. T. C. Wong. Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy. *IEEE Transaction on Biomedical Engineering*, 53(4):762–766, 2006.

[130] O. Debeir, P. V. Ham, R. Kiss, and C. Decaestecker. Tracking of migrating cells under phasecontrast videomicroscopy with combined mean-shift processes. *IEEE Transactions on Medical Imaging*, 24(6):697–711, 2005.

Publications

Publication I

M. Farhan, P. Ruusuvuori. M. Emmenlauer, P. Rämö, C. Dehio, and O. Yli-Harja, "Multi-scale Gaussian representation and outline-learning based cell image segmentation," *BMC Bioinformatics, 14(Suppl 10):S6, August 2013.*

Farhan et al. BMC Bioinformatics 2013, 14(Suppl 10):S6
http://www.biomedcentral.com/1471-2105/14/S10/S6

BMC Bioinformatics

RESEARCH Open Access

Multi-scale Gaussian representation and outline-learning based cell image segmentation

Muhammad Farhan[1*], Pekka Ruusuvuori[1], Mario Emmenlauer[2], Pauli Rämö[2], Christoph Dehio[2], Olli Yli-Harja[1]

From 10th International Workshop on Computational Systems Biology
Tampere, Finland. 10-12 June 2013

Abstract

Background: High-throughput genome-wide screening to study gene-specific functions, e.g. for drug discovery, demands fast automated image analysis methods to assist in unraveling the full potential of such studies. Image segmentation is typically at the forefront of such analysis as the performance of the subsequent steps, for example, cell classification, cell tracking etc., often relies on the results of segmentation.

Methods: We present a cell cytoplasm segmentation framework which first separates cell cytoplasm from image background using novel approach of image enhancement and coefficient of variation of multi-scale Gaussian scale-space representation. A novel outline-learning based classification method is developed using regularized logistic regression with embedded feature selection which classifies image pixels as outline/non-outline to give cytoplasm outlines. Refinement of the detected outlines to separate cells from each other is performed in a post-processing step where the nuclei segmentation is used as contextual information.

Results and conclusions: We evaluate the proposed segmentation methodology using two challenging test cases, presenting images with completely different characteristics, with cells of varying size, shape, texture and degrees of overlap. The feature selection and classification framework for outline detection produces very simple sparse models which use only a small subset of the large, generic feature set, that is, only 7 and 5 features for the two cases. Quantitative comparison of the results for the two test cases against state-of-the-art methods show that our methodology outperforms them with an increase of 4-9% in segmentation accuracy with maximum accuracy of 93%. Finally, the results obtained for diverse datasets demonstrate that our framework not only produces accurate segmentation but also generalizes well to different segmentation tasks.

Introduction

High-throughput screening used in drug design involves identification of genes which modulate a particular biomolecular pathway. RNA interference (RNAi), by decreasing the expression of particular genes in a cell culture, helps in identifying and analyzing the target gene functions in the cells by observing the cell behavior after gene knockdown [1-3]. Image analysis is at the center stage of such studies where cell cultures are imaged with automated fluorescent microscopy to study the cell behavior in knockdown as well as in normal condition. Genome-wide high-content siRNA screening involves studying the dynamics of gene expression in cellular functions for the whole genome and therefore yields hundreds of thousands of images making their manual analysis impractical [3]. Quantitative image analysis is needed for the identification, classification and quantification of the phenotypes which is also not possible through manual analysis [3,4]. Consequently, fast enough automated image analysis methods are needed to fulfill the potential of high-throughput system.

Segmentation of cells is typically at the core of the image analysis pipelines dealing with high-content genome-wide screening experiments [4,5]. This is generally the step which performs cell detection and further analysis, such as cell tracking and lineage reconstruction and cell classification, is based on the results of cell detection.

* Correspondence: muhammad.farhan@tut.fi
[1]Department of Signal Processing, Tampere University of Technology, 33720 Tampere, Finland
Full list of author information is available at the end of the article

However, in such experiments, segmentation is challenging due to presence of large number of phenotypes. Different cell phenotypes have different characteristics and appearances and, for some complex and heterogeneous cell cultures, it is difficult to build analysis capable of detecting all the phenotypes, potentially leading to the loss of some phenotypes. Accurate cell segmentation and detection is therefore essential for quantification of phenotypes.

One of the main challenges in cell segmentation is the cells touching and clustering together, forming a clump. Not only the cytoplasms form clumps but clustering of nuclei is also quite common. The latter problem has been tackled in our recent article [6]. The problem with cytoplasm region in general, and specifically with their clumps, is that they do not often have visible boundaries. Due to this reason, and also due to their irregular shapes, the methods typically in use for clump splitting often fail [7]. The other challenge often faced in cytoplasm segmentation is uneven and varying actin signal. Imaging aberrations cause actin signal to be saturated at some locations and to be too low on other locations for being regarded as part of the cell. This causes methods based on global image segmentation methods to fail. Another similar challenge that lies in cytoplasm segmentation is that the inside of the cells is inhomogeneous, consequently the intensity variations are large. Sometimes, part of the cell cytoplasm resembles the background and the methods solely based on image intensity are often found struggling in such situations [4]. However, if along with image intensity, other features locale to those regions are examined, the difference between background and cytoplasm could be highlighted. In addition to all this, uneven illumination and out of focus regions of the image also cause problems in getting accurate segmentation results.

Methods for cell cytoplasm segmentation available in literature can be mainly divided into two approaches: classic segmentation methods and deformable model-based methods. The former includes watershed transform, region growing, and mathematical morphology methods etc., see for example [8,9], whereas the latter comprises active contour [10], level set [11,12] and graph cut based methods [5]. Authors in [7] developed a method in which watershed algorithm with double thresholds is followed by splitting and merging of cellular regions based on quality metric obtained by correctly classified cells. Classification of cells is performed using a set of features with *a priori* information about the cells. In [13], enhancement of high intensity variations in the actin channel is performed by variance filtering. The enhanced image is then smoothed and thresholded using Otsu thresholding method. Subsequently, seeded watershed transform is applied which is restricted to the binary image of the cytoplasm. In another method [5], region growing algorithm and modified Otsu thresholding are used to extract the cytoplasm. Long and thin protrusions on spiky cells are extracted by scale-adaptive steerable filter. Finally, constraint factor graph cut-based active contour method and morphological algorithms are combined to separate tightly clustered cells.

In a method described in [4], the interaction between cells is modeled using a combination of both gradient and region information. Energy function is formulated based on an interaction model for segmenting tightly clustered cells. The energy function is then minimized using a multiphase level set method. Markov Random Fields (MRF) based graphical segmentation model yielding energy minimization problem is also applied to cell cytoplasm segmentation where graph cut method is used to obtain an exact MAP solution [14]. Similarly $\mathcal{P}^n$ Potts model, where functions of higher-order cliques of pixels are included into the traditional Potts model, combined with learning methods for defining the potential functions accounting for local texture information are used to segment live cell images in [15].

The problem with these methods is that they tend to produce over- and/or under-segmentation, for example, classic segmentation methods. Also, they are sometimes computationally-intensive and slow or they depend on schemes which require parameter initialization, and finding a good set of initial parameters for large heterogeneous dataset often requires user intervention which hinders development of automated analysis pipelines [16]. Moreover, when the cells are non-convex, as in our case, the methods available for segmentation of convex objects do not work, nor do the methods which are based on shape priors.

When cells clump together the cytoplasm outlines become invisible, however the intensity and other features along that part of the image are quite similar to the features of other cell outlines that are visible. Therefore, a segmentation methodology can be developed in which the outlines of the cell cytoplasm are learned by a supervised machine learning algorithm. There are methods in literature [17-20] which use the technique of learning edges for segmentation and object detection. However, all of them detect and model outlines which are distinct, where the outlines are basically used to detect objects or regions in the image utilizing shape information wherever available. In contrast, we need an outline detection technique which not only detects distinct outlines but is also capable of revealing outlines to separate objects of unknown shapes from each other.

In this paper we propose a supervised learning and classification-based cell cytoplasm segmentation methodology in which the outlines of the cell cytoplasm are learnt and detected. A multi-scale approach is used to get the cytoplasm/background segmentation and the detected outlines are overlaid to get the complete segmentation. The results

from the classification framework are fed to post-processing phase, where the methodology uses the nuclei segmentation [6] as contextual information to refine the segmentation results.

The rest of the paper is organized as follows: In the Methods section, we describe the proposed cell cytoplasm segmentation methodology. The obtained results are presented and discussed in Results and discussion section. The last section concludes the paper.

Methods

The proposed cell segmentation methodology involves three steps which are delineated by the block diagram in Figure 1. Firstly, images are passed through a pre-processing stage where most of the imaging aberrations are dealt with before applying multi-scale approach to separate cytoplasmic regions from the image background. Secondly, features are extracted from image pixels and a classifier is trained for classification of image pixels as either outline or non-outline to detect the cell outlines. Finally, a post-processing step is performed to refine the outlines so that they form a closed contour around each cytoplasm to get the individual cells segregated from each other. Implementation of the methods and additional information are available online https://sites.google.com/site/cellsegmentationhcs/.

Cell cytoplasm segmentation

The first step in our segmentation methodology is robust cytoplasm/background segmentation. As we mentioned earlier, there are many aberrations linked with high-throughput fluorescent microscopy imaging systems. Briefly described, the images typically have low contrast, with blurred regions around the image corners, varying signal strengths, inhomogeneous cell interiors and they also sometimes have uneven illumination. Generally, cytoplasm images appear to be most affected by these problems as far as their accurate segmentation is concerned.

Apart from these imaging related challenges, the other challenge that we face is posed by our dataset which includes cells with high phenotypic variability. Examples of challenging phenotypes are ruffles and spikes in cell boundary and other kinds of outline variations. A segmentation method robust enough to detect such fine details from the noisy and low contrast images is needed for

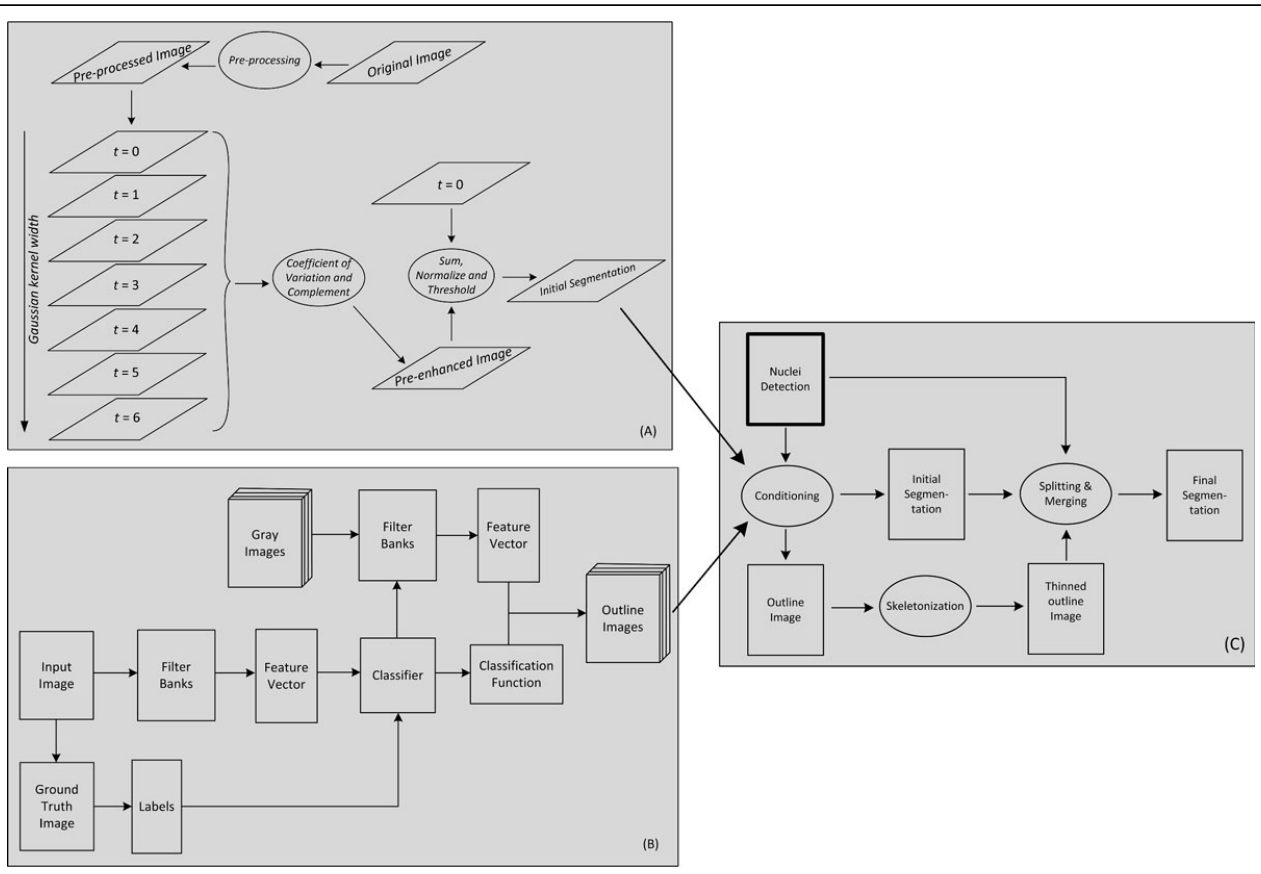


Figure 1 Block diagram of cytoplasm segmentation methodology. A block diagram showing the steps performed by the proposed cytoplasm segmentation methodology.

distinguishing different phenotypes. Our approach is to first apply enhancement and correction to the images before applying any segmentation method. Here, we use a cascade of three image and contrast enhancement filters for image pre-processing and a multi-scale approach for getting the desired initial cytoplasm/background segmentation. Block (A) in Figure 1 shows the steps performed in getting initial cytoplasm segmentation.

Image pre-processing

A cascade of image and contrast enhancement filters is used to preprocess the image to solve most of the above mentioned problems. First, contrast-limited adaptive histogram equalization [21] is applied to enhance the contrast of the image. The image is divided into 8×8 tiles and contrast of each tile is enhanced and the neighboring output tiles are combined using bilinear interpolation to avoid artifacts. In homogeneous regions of the image, over-saturation is avoided by clipping the high histogram peak occurring due to many pixels with similar intensity values. Then we applied opening by morphological reconstruction to the contrast enhanced image (mask) using a marker image. The marker image is created by eroding the mask image by a flat disc-shaped structuring element of radius of 5 pixels. The advantage of performing opening by reconstruction over conventional morphological opening is that, after opening, the topology of the cytoplasmic regions remains intact. It mainly smoothens out spurious high and low valued pixels and tackles the problem of uneven and varying actin signal. Finally, contrast of the image is adjusted once more by saturating 1% of the high and low intensity valued pixels. We will see that this is also beneficial for the image processing at the next stage. Figure 2(a) shows an original actin channel cytoplasm image and (b) the corresponding preprocessed image.

Multi-scale coefficient of variation based cytoplasm segmentation

After pre-processing the cytoplasm image, the initial cytoplasm/background segmentation is performed using our novel approach. Difference of Gaussian is a well-known technique used to enhance the edges in the image, especially the ones corrupted with noise [22]. On the other hand, for a stack of brightfield images, coefficient of variation has been found to be effective in contrast and details enhancement [23]. Our approach effectively combines the characteristics of these two approaches. It is based on coefficient of variation of the multi-scale Gaussian scale-space representation of the cytoplasm images to enhance the low contrast cytoplasmic regions. For an image $f(x, y)$, its Gaussian scale-space representation is a family of derived signals [24] given by

$$L(.,\ .;t^2) = g(.,\ .;t^2) * f(.,.);\ t \geq 0, \tag{1}$$

where

$$g(x,\ y;\ t^2) = \frac{1}{(2\pi t^2)} e^{-(x^2+y^2)/2t^2},$$

is a Gaussian kernel of increasing width t and * stands for the convolution operation. The parameter t is a parameter indicating the scale and at $t = 0$ the scale-space representation is the image $f(x, y)$ itself. For increasing value of t, L is an increasingly smoothed version of $f(x, y)$ with lesser details in the image. In our study, the scale-space representation is composed of seven images obtained at scales $t = [0, 1, 2, 3, 4, 5, 6]$ corresponding to the original image and their coefficient of variation image f_{COV} is given by

$$f_{COV}(x,\ y) = \frac{\sqrt{E[(L(.,\ .;t^2) - E[L(.,\ .;t^2)])^2]}}{E[L(.,\ .;t^2) + \varepsilon]}, \tag{2}$$

where $E[\cdot]$ is the expectation operator and $\varepsilon = 1$ is used to avoid probable outliers due to division by zero at pixel locations with zero intensity value. This leads to an image with higher values at image background and the cytoplasm outline pixels and relatively lower values for cytoplasmic regions of the image. Moreover, due to the standard deviation of stack of blurred images at different scales, it also enhances the edges and highlights the less bright spikes and ruffles of the cytoplasm. This also helps in differentiating the image background pixels from the less bright regions of the cytoplasm caused by intensity inhomogeneities. Adding the inverse of this image, after normalization, to the image $f(x, y)$ leads to an enhanced image $f_{enh}(x, y)$ with cytoplasm pixels clamped at a more higher value while background pixels at a relatively small value, that is,

$$f_{enh}(x,\ y) = f(x,\ y) + (2^b - 1 - f_{COV}(x,\ y)), \tag{3}$$

where b is the number of bits used to represent the image. This enhancement in image increases the difference between the darkest cytoplasm pixel and the brightest background pixel and a simple intensity threshold-based method such as Otsu segmentation [25] is able to give the desired cytoplasm/background segmentation. Figure 2(b) shows a gray-scale pre-processed cytoplasm image, 2(c) the coefficient of variation image and 2(d) the resulting image with cytoplasm/background segmentation. From the figure, it is quite evident that our method is able to detect the cytoplasmic regions correctly despite the presence of intensity inhomogeneities.

Classification-based cell cytoplasm outline detection

The cytoplasm segmentation obtained in the previous step still has cytoplasms of different cells touching each other. This is the step in which we detect the cytoplasm

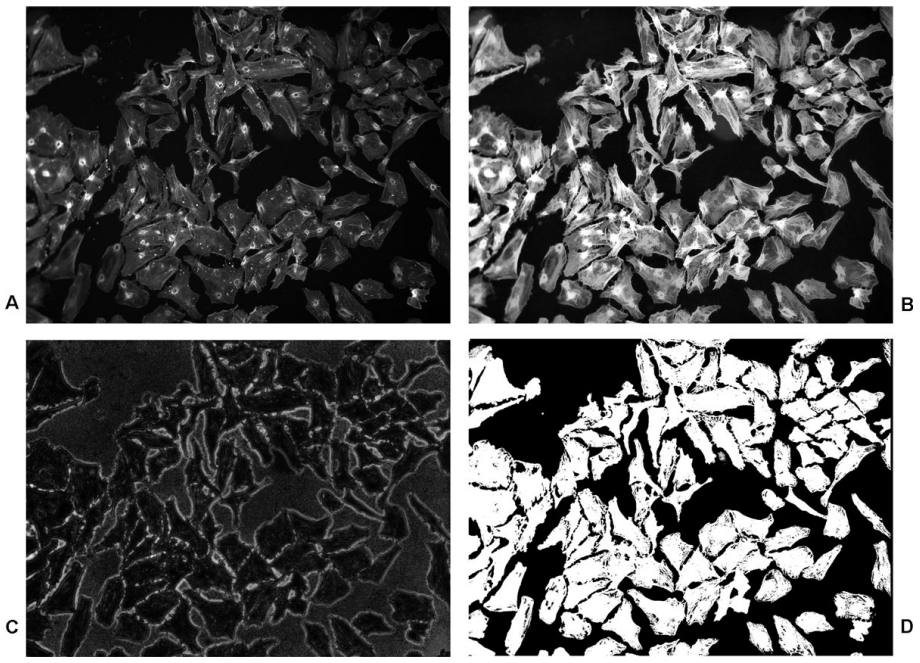


Figure 2 Image pre-processing and cytoplasm/background segmentation. Image pre-processing and cytoplasm/background segmentation. (a) An actin-channel cell microscopy image showing the cell cytoplasm and (b) the result of pre-processing. (c) The coefficient of variation image of scale-space representation and (d) the resulting cytoplasm/background segmentation. The size of the image is 1040×1392 pixels.

outlines and apply them to the result of the previous step for getting the whole cell segmentation. As we mentioned earlier, even if the cytoplasm outlines are invisible, especially in the regions where cytoplasms clump, the intensity and other features of the pixels with underlying outlines still closely match the features of outline pixels that are clearly visible. This leads us to an approach in which a classifier is trained to classify a pixel as either outline or non-outline based on the set of local features extracted from the image pixels.

A large set of generic pixel-level features is extracted from the training image using a set of filter banks, see Table 1. Using these features and training labels obtained from manually outlined image(s), a classifier is designed utilizing sparse logistic regression classification framework which has feature selection property inherent to it. For any test image, only the features selected by the classifier are extracted and using the designed classifier the image pixels are classified as outline/non-outline pixels, see block (B) in Figure 1.

Extraction of features

The complexity and accuracy of a classifier depends upon the number and distinguishing nature of the features used for classifier design. Selection of the most informative features from a list of candidate features reduces the model complexity yet it needs to be performed such that the model yields high classification accuracy. Sparse model using only a subset of the available features allows us to keep the initial feature set large with as many general and redundant features as desired. Moreover, the benefit of using large and general rather than small and problem-specific feature set is that the framework generalizes to other similar classification problems. Hence, we employ an exhaustive set of generic linear and non-linear features knowing our feature selection technique has been

Table 1 Filtering operations and the filter parameters for computing pixel-level features from training images.

Operation (Feature)	Parameter	Values	Total
Gaussian low pass	kernel width σ	3:2:49	24
Integrated pixel intensity	kernel size	3:2:9	04
Laplacian of Gaussian	kernel width σ	3:2:49	24
Difference of Gaussian	kernel width σ		05
Morphological top-hat	kernel size	3:2:49	24
Morphological bottom-hat	kernel size	3:2:49	24
Local binary pattern and contrast	(quantization, radius)	(8,1)	02
Variance	kernel size	3:2:49	24
Order statistics (Min., Med., Max.)	kernel size	3:2:7	09
Haralick (13-features)	kernel size	5:2:15	78
Gabor filter	kernel size, freq. f, orientation θ	5:2:15, 1/4:1/4:3/4, 0:π/4:3π/4	72
Total number of features.			290

successfully used for building sparse classification models in similar use cases [26].

In our study, pixel-level features are extracted from 2D cytoplasm images by applying a large set of filters on them, both in spatial and transform domain, with varying parameters. In [26], the authors use a large generic set of intensity-based features along with textural feature such as local binary pattern (LBP) [27] for image segmentation. Our cytoplasm images possess interesting texture characteristics which might be useful in classification of image pixels. Therefore, in addition to the local binary patterns and other intensity features used in [26], we also incorporate texture features such as the ones obtained from Gabor filters [28] and Haralick [29] features in our classifier design. The feature set comprises general intensity, edge, texture (scale and orientation) based local features which are computed in the pixel neighborhoods using filters with varying kernel sizes. Table 1 lists all the features that are computed for the training images.

Design of classifier incorporating feature selection

High-dimensionality of the observations leads to the risk of over-fitting at the cost of generalization of the solution and reduction of feature space is desired. However, selection of the most informative features from a feature set for modeling data characteristics has always been problematic. In case of multiple linear regression modeling, regularization is a process which adds a penalty term to the least square prediction error to shrink the magnitude of model coefficients towards zero. Thus a sparse solution with only few non-zero coefficients is obtained and feature selection is performed automatically. Least absolute shrinkage and selection operator (LASSO) [30] is a technique which penalizes the error function using l_1-norm of coefficient vector along with a regularization parameter $\lambda > 0$ which controls the sparsity of the solutions. This is another characteristic of this framework, that is, its provision of a set of solutions which usually has increasing sparsity for an increasing value of λ. The advantage in it is that it helps in choosing a solution with as many features desired with little or no major change in the classification result, that is, a solution with a small trade-off between accuracy and model sparsity/complexity.

Using such framework, a classifier with sparse model is designed by taking the advantage of logistic function to describe the class probability $p(o_i|\mathbf{x}_i)$ of pixel i belonging to outline by

$$p(o_i|\mathbf{x}_i) = \frac{1}{1 + e^{(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}}, \tag{4}$$

where o_i represent the class "outline" and probability for class "non-outline" n_i is given by $p(n_i|\mathbf{x}_i) = 1 - p(o_i|\mathbf{x}_i)$, $\mathbf{x}_i \in \mathbb{R}^p$ denotes the feature vector of the i^{th} pixel and $(\beta_0, \boldsymbol{\beta})$ is the coefficient vector which is estimated by maximizing the penalized log-likelihood given by

$$\sum_{i=1}^{N} \{\log p(o_i|\mathbf{x}_i) + \log(1 - p(o_i|\mathbf{x}_i))\} - \lambda ||\boldsymbol{\beta}||_1, \tag{5}$$

whose quadratic approximation gives rise to an equivalent penalized iteratively re-weighted least squares problem that can be solved by coordinate descent algorithm [31].

Training and classification

In order to perform training and classification, manually created benchmark images with cytoplasm outlines are used. We have a set of training samples, around 550 cells (5 images) and 1250 cells (16 images) for *Test Case I* and *Test Case II*, respectively, segmented manually by expert biologists, see details regarding image acquisition in later section. It is worth-mentioning that, while choosing the images for benchmarking, the criteria was to pick those images which contain most of the image area covered with cells and also the chosen images present one of those cases which are the most challenging as far as getting accurate segmentation is concerned. Since all the images are 1040×1392 (*Test Case I*) and 400×400 (*Test Case II*) in size, even the pixels of a single image are sufficient enough to train the classifier, especially the classifier of our type which is capable of dealing with even $P \gg N$ cases. Therefore, one of the images is used solely for training of the classifier while the rest of the images are used for evaluating the classifier. This way we made sure not to use the same data for both training and testing.

For training, 500 positive (outline) and 500 negative (non-outline) samples are picked at random from 1447680 or 160000 samples in the benchmarked image of cytoplasm outlines. For these 1000 samples, all the 290 features listed in Table 1 are extracted from the corresponding cytoplasm image. This training data of 1000×290 feature vector along with 1000×1 target labels is input to the regularized logistic regression classifier. For testing, only the selected features are calculated for every pixel in the test images to be used with the selected model for outline classification.

In order to estimate the optimal classifier model coefficients, 10-fold cross-validation is performed on the training data to estimate the prediction error of all the solutions obtained for different values of regularization parameter λ. The solution which gives the minimum prediction error is generally chosen, however, it can be left to the discretion of the designer to pick an even more sparse solution with little or no major impact on the final classification results. In our case, we observed that models within one standard error of the mean cross-validation error do

not change the classifier output significantly. Finally, the selected model for the classifier gives the posterior probability values for the pixels in the test image which is used directly to find the class label (outline/non-outline) for every pixel.

Post-processing

Post-processing of the classifier outputs is generally a complementary part of any classification framework. One of the techniques used for post processing exploits the contextual information obtained either from the targeted patterns, which, in our case is cytoplasm images, or from some other source related to them. The classifier that we obtained to classify image pixels as outline/non-outline gives accurate yet coarse results. The coarseness mainly comes from the fact that sometimes the pixels interior to the cytoplasm are given the outline labels due to similarity of their features with outline pixels which was actually caused by varying and inhomogeneous actin signal. Moreover, due to binary outputs, that is, the threshold probability value of 0.5, the classifier tends to give thick outlines because many pixels close to the actual outline have similar features with little variations among them. Also, again due to varying signal strength or due to noise, quite often the detected outlines are non-connected, whereas, the desired solution is to have closed contour outlines for cytoplasms. Therefore, we need to refine the classifier output and transform it in such a way that we get single-pixel length closed outline contours.

In eukaryotic cells, nucleus is the main indicator of a cell. We have the DNA-channel nuclei images which provide a solid basis to find the individual cells, or to detect individual cell cytoplasm outlines in the actin-channel cytoplasm images. In cell images, nucleus is generally located at the central portion of the cell. Most importantly, we can certainly assume that the pixels occupied by the nucleus can never be occupied by the cell outlines. Therefore, nuclei images provide contextual information for post-processing of the classifier output. Mainly, they are used to filter out the misclassified outline pixels lying inside the cell. In the same context, they are also used to refine the result of initial segmentation to fill underlying small holes occurring due to intensity inhomogeneities. This image is then inverted and unified with the filtered outline image to further strengthen the outlines.

Once the outlines are filtered, their thinning is performed by morphological skeletonization to get single-pixel length outline contour. Skeletonization is preferred over morphological thinning since it gives not only accurate contour in terms of its location but it also gives non-connected branches wherever available. These branches occur either due to discontinuous outlines or due to some noisy structures in the original cytoplasm images, and help in getting closed contour outlines. Decision on whether to join these non-connected branches or not is taken on the basis of object correspondence at the nuclei and cytoplasm level. In order to find the correspondence, the thinned outlines are applied on the initially segmented images to get the first-stage cytoplasm segmentation. Due to false positives and false negatives in the outlines classification we get over- and under-segmentation. To deal with this, nuclei images are used to perform an additional step of splitting and merging.

In the splitting and merging step, firstly, nuclei image is used to morphologically reconstruct the first-stage cytoplasm segmentation image. This separates objects or cytoplasmic regions with and without a corresponding nucleus. The latter ones are saved to be merged in a later part of this step. In the former case, we have two types of correspondences: one-to-one correspondence between cytoplasm and nucleus and one-to-many correspondence between cytoplasm and nuclei. In the former case, there is one nucleus for every cytoplasm which is often the case in our images as there are very few multinuclear cell phenotypes. Morphological closing is applied to such objects to smoothen inside of cytoplasm and to remove any non-connected branches occurring due to noise or intensity inhomogeneities.

In the case of one-to-many correspondence, the respective non-connected branches in outline are extracted and dilated to close in the gaps. Skeletonization and morphological reconstruction are applied again to split the regions into nucleus-bearing regions and non-nucleus-bearing regions. It is worth-mentioning that no extra splitting approach is used in order to get one cytoplasmic region per nucleic region. The reason is that the nuclei used for finding correspondence are themselves found to be affected by over-splitting and an attempt to forcefully split a cytoplasmic region despite the absence of outline would result in cytoplasm over-segmentation translated from nuclei over-segmentation. Moreover, our approach also helps in retaining the morphology of the multinuclear cell phenotypes.

Finally, region merging is performed to merge all the non-nucleus-bearing regions resulting from the previous step with the separated nucleus-bearing cytoplasmic regions. Candidates for merging are obtained by dilating the to-be-merged regions and finding the overlapping regions in the nucleus-bearing cytoplasmic regions. Since the cells in our image set are mostly convex, therefore, in the case of more than one candidates, the one which gives the largest solidity is chosen. The process is repeated for a couple of more iterations so that regions that do not have an overlapping cytoplasm initially, due to being away from a cytoplasmic region, may have one now due to their adjacent regions being merged with a cytoplasmic region in

the previous iteration. In the end, morphological operations are performed to remove h-connectivity as well as 8-connectivity of the objects and to fill small holes in them. Block (C) in Figure 1 outlines the steps performed in post-processing to get the final segmentation result. Figure 3 shows the results of outline detection and post-processing for the segmented image of Figure 2.

Results and discussion

To study and analyze the performance of our segmentation methodology, we test it against two challenging test cases. Both of them consist of image sets of different cell types with cells of varying size, shape, texture and degree of overlap. The first case is challenging in the sense that it contains images with high cell density with large variation in the shape as well as in size of the cells. The second test case is more of a validation case because it not only contains images from publicly available dataset with ground truth benchmarking, but it also presents an altogether different set of images from the first test case. This enables testing the generalization of our framework. The challenging aspect of the second case, similarly as for the first case, is that the cells are such tightly clustered with virtually no indiscernible boundaries that even accurate manual segmentation is sometimes impossible. Moreover, in both the cases, the extensive variation in signal strength, intensity inhomogeneity and low contrast make the segmentation task even more challenging.

Image acquisition

The details about the experimental settings to perform image acquisition for compiling the dataset for *Test Case I* and *Test Case II* are given below.

Test Case I

Experiments were conducted in a 384-well plate format imaging *HeLa CCL-2 ATCC* cells using Molecular Devices ImageXpress microscopes (10× objective; 9 sites per well, Channels DAPI: DNA, GFP: pathogen, RFP: actin) with robotic plate handling. The objective was 10X S Fluor. Image binning was not used. Gain was set to low (Gain1). Laser-based focusing was enabled and image-based focusing was disabled. The dynamic range was set to 12 Bit Range. Z-Offset for Focus was selected manually and *AutoExpose* was used to get a good exposure time. Manual correction of the exposure time was applied to ensure a good dynamic range with low overexposure, when necessary. The size of each image is 1040×1392 pixels. Manual benchmark creation was performed by biologists where cell cytoplasm outlines are drawn. Due to the presence of multinuclear phenotypes, there are few cases of multiple nucleus per cytoplasm. Five images containing around 550 cells were taken which were representative of most of the problematic cases not solved well by a widely used method from [32].

Test Case II

In this test case we use images of *Drosophila melanogaster Kc167* cells which were stained for DNA (nuclei)

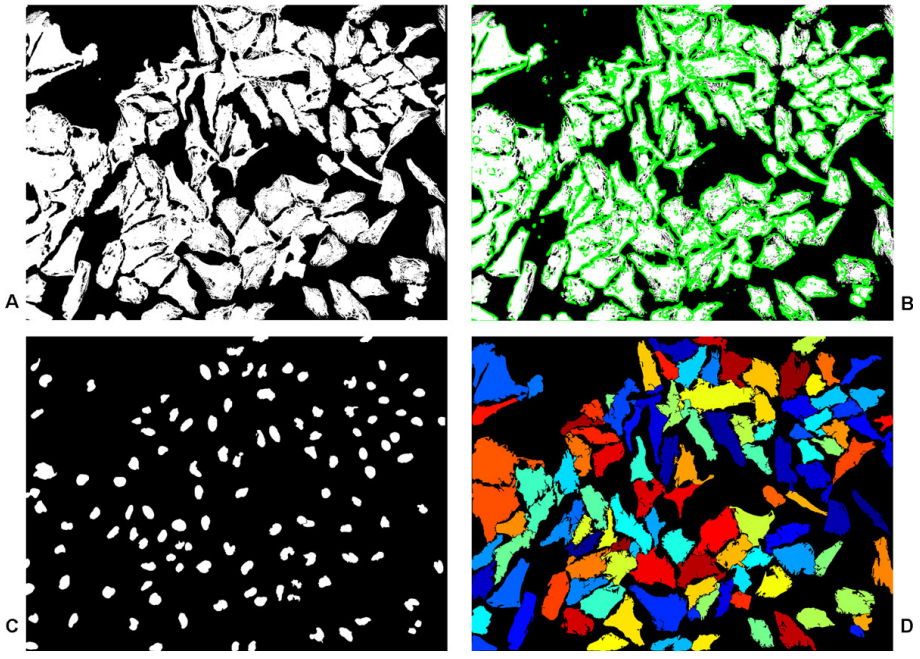


Figure 3 Outline detection and post-processing. Outline detection and post-processing. (a) An image after initial segmentation. (b) Resulting outlines (green) from classification of image pixels into outline/non-outline pixels. (c) Corresponding DNA-channel nuclei image, segmentation obtained from method in [6]. (d) Final segmented image after post-processing. The size of the image is 1040×1392 pixels.

and actin (cytoplasm). "Images were acquired using a motorized Zeiss Axioplan 2 and a Axiocam MRm camera, and are provided courtesy of the laboratory of David Sabatini at the Whitehead Institute for Biomedical Research. First, nuclei were outlined by hand. The nuclear outlines were overlaid on the cell images, and one cell per nucleus was outlined" [33]. There are 16 images in the dataset with size 400×400, 450×450 and 512×512 pixels, containing cells of around 25 pixels in diameter with an average of 80 cells per image. The motivation for using this image set primarily comes from its public availability and benchmarking. Also, these images provide challenging segmentation tasks which have also been worked upon previously, such as in [16,34,35]. This helps in examining the proposed method in comparison to the results obtained from these state-of-the-art methods.

Segmentation quality metrics

To evaluate the accuracy of our segmentation method and to quantitatively compare it with other methods, we obtained performance metrics at two different levels: pixel level (cytoplasm image) and object level (both nucleus and cytoplasm images). The performance metric that we used is F-measure (*FM*), like we did in [6,36], which is the harmonic mean of Precision (*PR*) and Recall (*RC*) and is given by

$$FM = \frac{2}{\frac{1}{PR} + \frac{1}{RC}}, \tag{6}$$

where

$$PR = \frac{TP}{(TP + FP)} \text{ and } RC = \frac{TP}{(TP + FN)}, \tag{7}$$

where *TP*, *FP* and *FN* are true positive, false positive and false negative, respectively, with respect to the benchmarked images. The higher the rate of true values, the lower the rate of false values and the higher would be the segmentation accuracy.

Pixel-level measures give an insight into how accurate the obtained segmentation is, in terms of correspondence between cells in segmented image and benchmarked image. For each cell in the benchmarked image, based on maximum overlap, a corresponding cell was found in the segmented image. *TP*, *FP* and *FN* values were obtained at pixel-level and *FM* value was obtained. In order for correspondence to be true, a threshold value of $FM_{th} = 0.6$ was used as it was used in [16]. Once an object correspondence is found, the object was removed from the segmented image and was not considered for any other object in the benchmarked image. In this way, only one-to-one (*TP*), one-to-none (*FN*) or none-to-one (*FP*) correspondence was obtained between the benchmarked image and the segmented image. This also accounted for the object-level measure for cytoplasms, that is, every one-to-one correspondence meant an increase in cell count. Object-level measures for the nuclei were also obtained in a similar way to get the nuclei count.

It is worth-mentioning that while finding correspondence for cytoplasms, the nuclei image was not used at all. The reason is that an over-splitting at nuclei level may not always cause over-splitting at cytoplasm level due to true absence of outline. Therefore, using nuclei for finding correspondence may result in wrong quantitative measures.

Nuclei segmentation

In both cases, nuclei segmentation was obtained by using our framework presented in [6]. However, in that framework we used graph cut segmentation method from [37] which can be replaced with the initial segmentation method proposed here for cytoplasm segmentation. From the results, it has been observed that although the nuclei segmentation framework with our proposed initial segmentation gives less smoother result than the framework with graph cut segmentation but when compared quantitatively it was able to reduce twice as many false negatives as it increases false positives. The reason is that our initial segmentation method was found to be better in detecting objects in low contrast with varying signal strength than graph cut method, even though the applied pre-processing was the same. Although, the final F-measure value was almost similar in either case, the decrease in false negative meant an increase in cytoplasm detection, whereas, a false positive might not be as costly since nuclei image is not affecting the splitting of cytoplasm regions as long as there is no underlying outline detected. For *Test Case II*, we replaced the graph cut-based initial segmentation of the framework in [6] with the initial segmentation method proposed here. As the magnification of these images is different from our images, that is, they have lesser pixels per nucleus, the set of values used for scale needs to decrease in order to avoid objects from getting connected due to larger kernel width. Therefore, Gaussian filtering was performed with smaller kernel width. Hence, the scale-space representation was composed of 7 images obtained at scales $t = [0, 0.5, 1, 1.5, 2, 2.5, 3]$ corresponding to the original image to get the initial segmentation as described in cell cytoplasm segmentation subsection.

Implementation details

In this subsection, we describe the procedure and the implementation details of the methodology for obtaining the results. In order to get the quantitative measures for evaluation, we applied our segmentation methodology

on the two image sets from the two test cases. First, we obtained nuclei segmentation in the way described in the previous subsection and the values of 600 and 100 were used for allowed minimum area of a nucleus for *Test Case I* and *Test Case II* respectively. Then, cytoplasm/background segmentation was obtained as mentioned in cell cytoplasm segmentation subsection. Finally, the outline/non-outline classifier design gave a sparse model with only eight non-zero coefficients for the *Test Case I* and linear model in denominator of Equation 4 turned out to be

$$\begin{aligned}\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta} = {} & 0.2415 - 44.998 * f_1 + 0.010 * f_2 - 0.006 * f_3 \\ & -0.009 * f_4 + 0.068 * f_5 - 0.207 * f_6 - 0.544 * f_7\end{aligned} \quad (8)$$

where $f_1 = VAR_{3\times3}$ stands for variance, $f_2 = MIN_{7\times7}$ for minimum, $f_3 = f1/4th0_{5\times5}$, $f_4 = f1/4th3pi/4_{5\times5}$ for Gabor filtering frequency and orientation, $f_5 = ASM_{5\times5}$ for angularSecondMoment, $f_6 = IMOC2_{7\times7}$ and $f_7 = IMOC2_{9\times9}$ for informationMeasureOfCorrelation2, see [29] for details. The subscript $x \times y$ stands for the respective kernel sizes. On the other hand, for the *Test Case II*, the classifier design gave a sparse model with only six non-zero coefficients and linear model in denominator of Equation 4 turned out to be

$$\begin{aligned}\beta_0 + \mathbf{x}_i^T\boldsymbol{\beta} = {} & 1.120 - 0.0165 * f_1 - 0.1790 * f_2 - 0.360 * f_3 \\ & -1.54 * f_4 + 0.1908 * f_5\end{aligned} \quad (9)$$

where $f_1 = ENT_{5\times5}$ stands for entropy, $f_2 = DOE_{7\times7}$ for differenceOfEntropy, $f_3 = IMOC2_{7\times7}$ and $f_4 = IMOC2_{9\times9}$ for informationMeasureOfCorrelation2 and $f_5 = ASM_{9\times9}$ for angularSecondMoment, see [29] for details. Again, the subscript $x \times y$ stands for the respective kernel sizes. Then, for each of the test images, feature vector of size 1447680×7 for *Test Case I* and 160000×5 for *Test Case II* were calculated and input to the above models to get the class probabilities using Equation 4. The probabilities were thresholded with threshold value of 0.5 to get outline/non-outline pixels. Finally, post-processing step was performed to get the segmentation done. Figure 4 presents a visual representation of the features used by classifiers given in (8) and (9).

Results and discussion

Quantitative values from the resulting images were obtained as described earlier in this section and are given in Table 2 and Table 3 for *Test Case I* and *Test Case II* respectively.

For the *Test Case I*, we have nuclei and cytoplasm segmentation results obtained from CellProfiler 1.0 (CP) implementation [32]. Table 2 also lists the values obtained from them. As we discussed about nuclei segmentation in [6], CP gives low value for *FN* , but at the expense of high value for *FP* . This high value of *FP* at nuclei level got translated into an even higher value at the cytoplasm level. This is because cytoplasm segmentation was purely based on nuclei segmentation and, effectively, one cytoplasmic region was found for every nucleic region. This difference in values for nuclei and cytoplasm segmentation is more due to FM_{th} value of 0.6 for cytoplasm detection. Since every over-splitting at nuclei level leads to over-splitting of cytoplasm which, most of the time, disqualifies all the cytoplasmic regions corresponding to an over-split nucleus. This is also evident from Table 2 that *FP* for cytoplasm became almost twice of *FP* for nuclei and those extra *FP* also affect the *FN* directly. Finally, the value of *FM* for CP cytoplasm segmentation came out to be 0.84.

As we mentioned earlier, our proposed cytoplasm segmentation mainly needs a low *FN* for nuclei segmentation because, due to cytoplasm-nuclei correspondence-based segmentation, cytoplasms for which nuclei are not detected are merged with other cytoplasms. Although, the *FM* values for CP implementation and our nuclei segmentation do not differ much, the detection error *FP* + *FN* for our method was 21, which is less than half as compared to 49 for CP implementation.

In the light of the discussion in the previous paragraph, forced splitting for obtaining one cytoplasm per every detected nuclei did not seem beneficial. However, the *FP* for our cytoplasm segmentation was still found to be twice as much as for nuclei segmentation. The reason is that objects that do not get split into constituent cells were no longer able to correspond to even a single object in benchmarked image because of the constraint of FM_{th}. Moreover, the consequence of avoiding forced splitting was an increased value for *FN* as some clumped cells did not get detected. A worth-mentioning point is that since the value of *FP* for our nuclei segmentation was low, forced splitting might still have resulted in a similar value of *FP* that we obtained without doing so, but that would have given a much lower value for *FN*. However, the main reason behind not using forced splitting was that we want to retain multinuclear cell phenotypes. The overall segmentation from the proposed method confirms that it outperforms the method from CP with a 9% increase in *FM* value. Another measure that we obtained is the mean value of *FM* for all the correctly detected cytoplasms and it was 0.85 for the proposed method against 0.81 for CP implementation. This also shows how well the cytoplasms correspond among the benchmarked images and our segmented images. Figure 5 presents the segmentation results from the proposed method for qualitative evaluation.

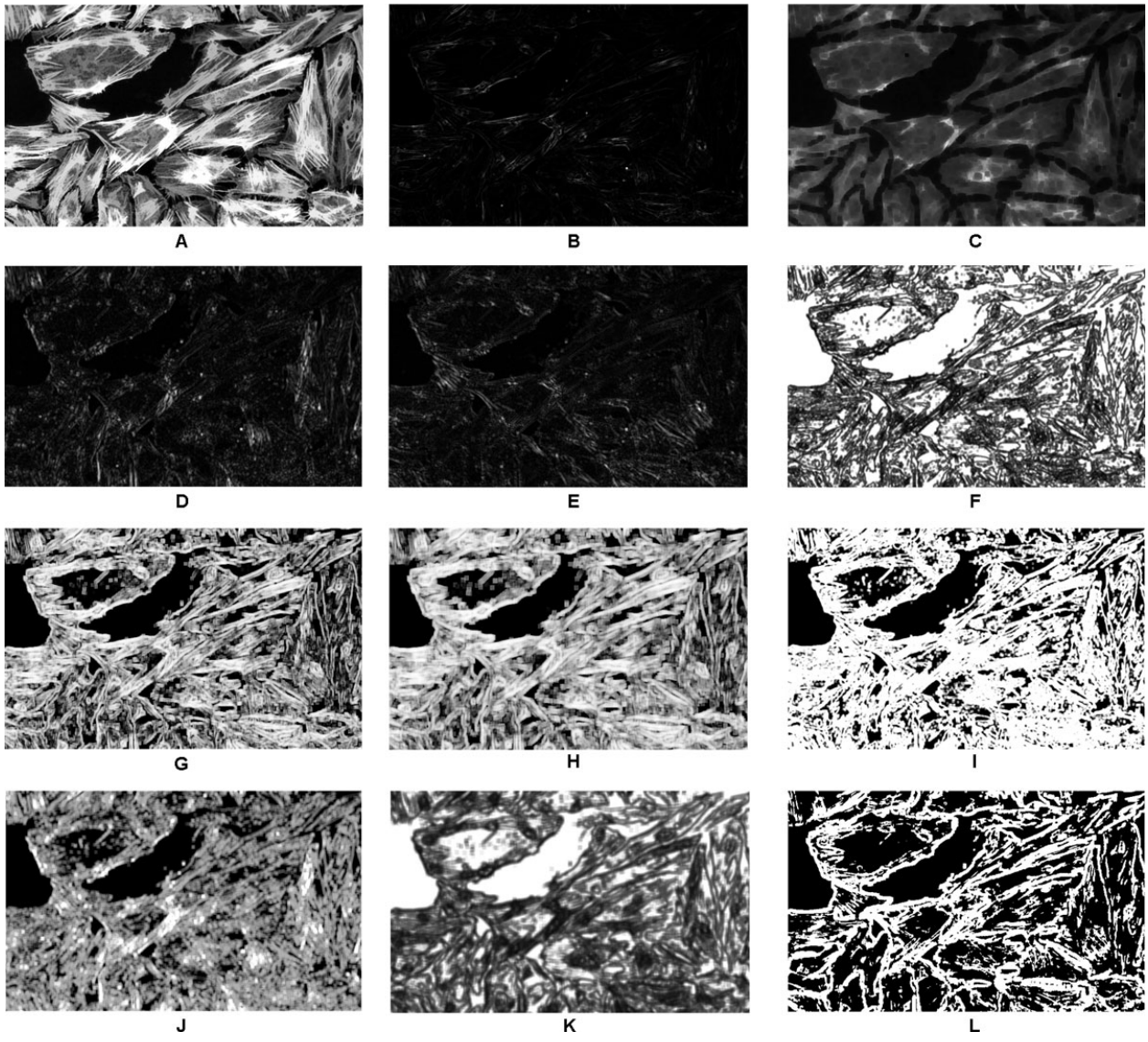


Figure 4 Visual representation of features used by classifiers. Visual representation of features used by classifiers. (a) A pre-processed image, (b) $VAR_{3\times3}$, (c) $MIN_{7\times7}$, (d) $f1/4th0_{5\times5}$, (e) $f1/4th3pi/4_{5\times5}$, (f) $ASM_{5\times5}$, (g) $IMOC2_{7\times7}$, (h) $IMOC2_{9\times9}$, (i) $ENT_{5\times5}$, (j) $DOE_{7\times7}$, (k) $ASM_{9\times9}$, and (l) outlines obtained from thresholding the output of classifier. The size of the images is 700×430 pixels.

The same images of *Test Case II* were used for performance evaluation of the cell nuclei and cytoplasm joint segmentation presented in [16]. Comparing the given values of *TP*, *FP*, and *FN* with our obtained values for cytoplasm segmentation, it can be said that we got similar or slightly improved results. However, it is difficult to say whether the difference has any significance. Moreover, the *FM* value from our method for nuclei detection is 0.95 as compared to the *FM* value of 0.80 reported in [16]. This suggests that our method outperforms a recently proposed

Table 2 Quantitative values obtained from nuclei and cytoplasm segmentation for Test Case I (See text for abbreviations).

Level (Method)	TP	FP	FN	PR	RC	FM
Nuclei ([6])	458	11	10	0.97	0.97	0.97
Nuclei (CP [32])	466	47	2	0.91	0.99	0.95
Cytoplasm (proposed)	424	23	42	0.95	0.91	0.93
Cytoplasm (CP [32])	409	103	57	0.80	0.88	0.84

Table 3 Quantitative values obtained from our segmentation method for Test Case II (See text for abbreviations).

Level (Method)	TP	FP	FN	PR	RC	FM
Nuclei ([6])	76	4	3	0.95	0.96	0.96
Cytoplasm (proposed)	70	9	9	0.89	0.89	0.89

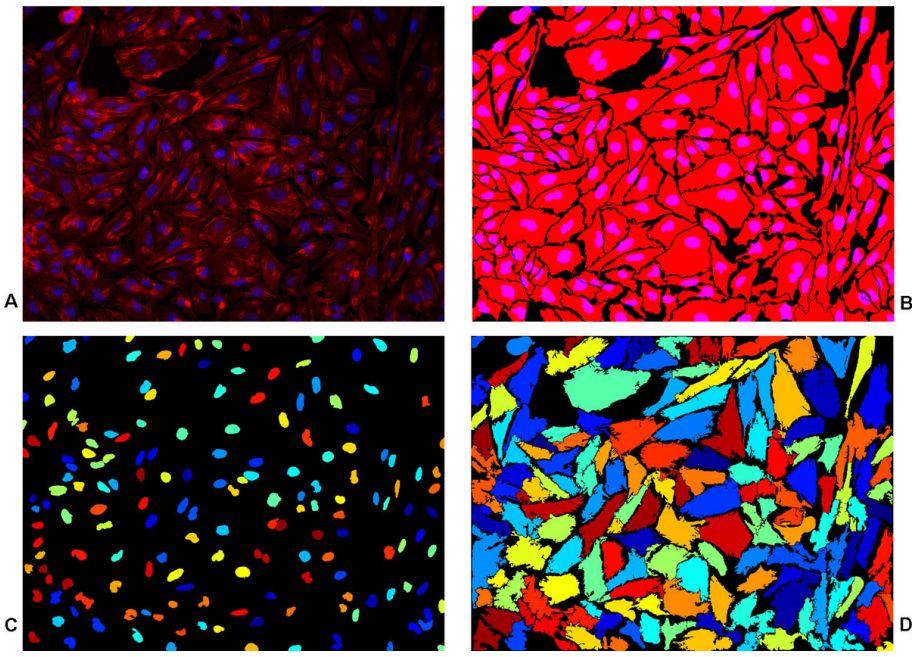


Figure 5 Cell cytoplasm segmentation for Test Case I. Cell cytoplasm segmentation for *Test Case I*. (a) A merged cytoplasm (Red)/nuclei (Blue) channel image, (b) benchmark segmentation from biologists, (c) nuclei segmentation from [6] and (d) the result of proposed segmentation. The size of the image is 1040×1392 pixels.

method which was also reported to be computationally quite expensive. Figure 6 shows the results of the proposed method for two images from *Test Case II*.

Finally, it is evident from the obtained qualitative as well as quantitative results for both the test cases that the proposed method was able to produce accurate results, see Table 2, 3 and Figure 5, Figure 6. Moreover, considering that both the test cases provide completely different set of images with different challenges, the obtained results also demonstrate the generic nature of our framework. In the end, it is worth-mentioning that even though the method uses manually outlined images for training the classifier, it does not depend on user-defined parameters for segmentation.

Conclusions

In this article we present a novel approach for cell segmentation. The proposed method uses a new combination of pre-processing methods for enhancing the contrast of cell cytoplasm and especially their boundaries by applying coefficient of variation for a multi-scale Gaussian representation of the input image. The enhanced image is used as a basis of feature extraction process, where filtering, texture operations and other generic descriptors are applied for building a large set of features to be used for building a classifier model for cell outline detection. By applying the logistic regression classifier, known to produce sparse models where only a subset of the initial features are used, a rather simple model with a small set of features is obtained, making the classification process computationally feasible. Finally, in post-processing phase, cell nuclei segmentation is used to aid the construction of final cell outlines from the classification output.

In order to validate the segmentation method, we used two image sets with different characteristics. The quantitative results confirm that the method performs consistently for the two datasets and when compared to a widely used method and values presented in literature, it can be concluded that our results are very promising; either improving or matching the results of earlier presented methods.

In conclusion, we expect that learning based methods may be useful in challenging segmentation tasks, such as in high content screening where low contrast cells should be accurately segmented in order to maintain high accuracy among challenging phenotypes. The labeled training samples, in this context: manually outlined cells in a set of images, is a fundamental requirement for using a supervised segmentation method. In high content screening the amount of image data is huge and since also the validation is in most cases done against manually segmented images, we feel that the gain in performance should justify the task of creating the training data.

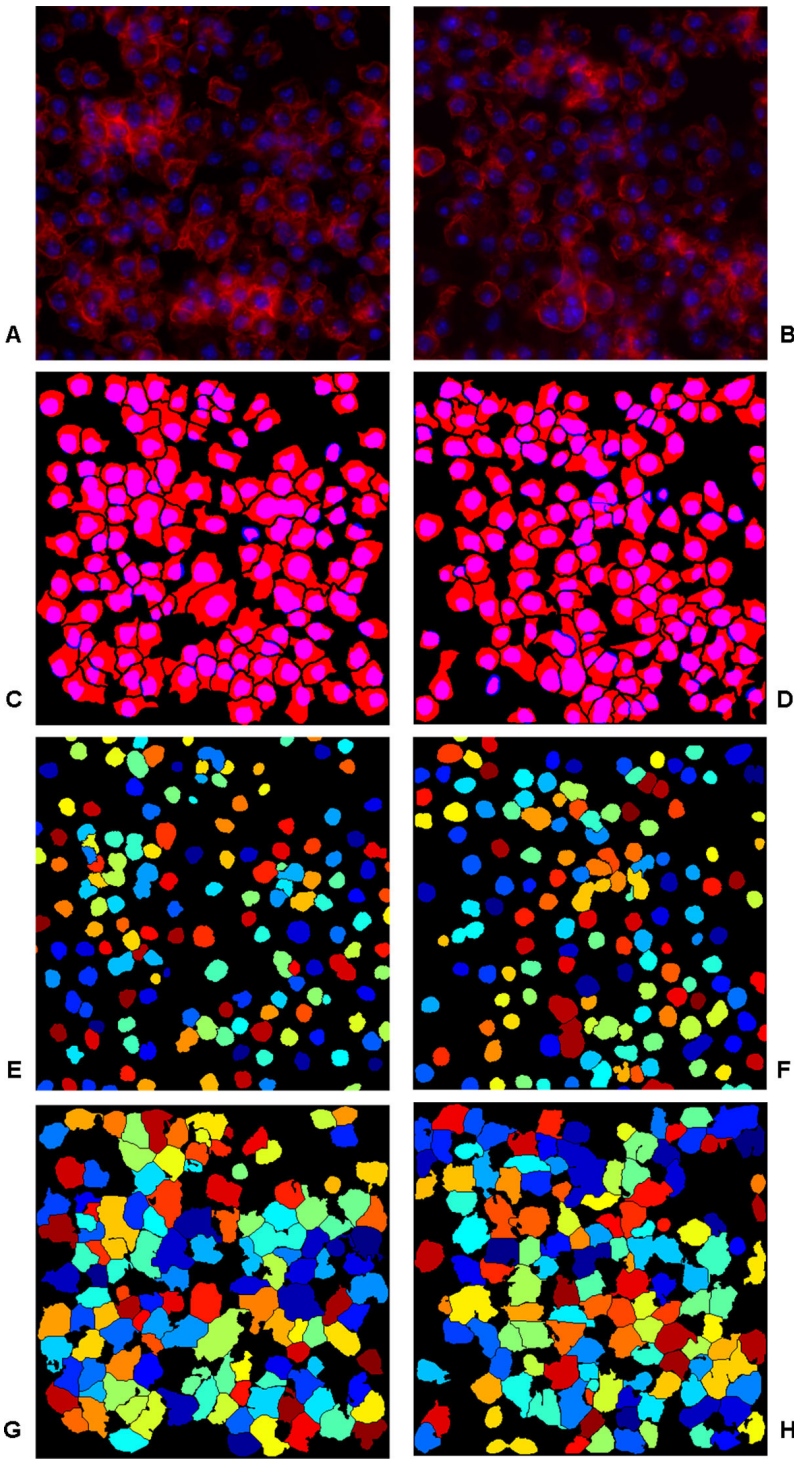


Figure 6 Cell cytoplasm segmentation for Test Case II. Cell cytoplasm segmentation for *Test Case II*. (a)-(b) Two merged cytoplasm (Red)/ nuclei (Blue) channel images, (c)-(d) benchmark segmentation, (e)-(f) nuclei segmentation from [6] and (g)-(h) the results of proposed segmentation. The size of the images is 450×450 pixels.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
Muhammad Farhan carried out the study, developed and implemented the methodology and wrote the manuscript. Pekka Ruusuvuori conceived of the study, coordinated algorithm design and computational experiments, and revised the manuscript. Mario Emmenlauer and Pauli Rämö participated in the design of the study and revised the manuscript. Christoph Dehio and Olli Yli-Harja participated in design of the study and coordination. All authors read and approved the final manuscript.

Acknowledgements
We acknowledge the financial support from TISE graduate school and Nokia Foundation (MF), Academy of Finland project #140052 (Pekka R), and grant 51RT-0-126008 (InfectX) in the frame of SystemsX.ch, the Swiss Initiative for Systems Biology (to Christoph D). We are also very grateful to the biologists at Biozentrum, Dr. Simone Eicher and Dr. Houchaima Ben Tekaya, for providing benchmark images of cell cytoplasm outlines.

Declarations
The funding for publication of the article comes from the aforementioned projects.
This article has been published as part of BMC Bioinformatics Volume 14 Supplement 10, 2013: Selected articles from the 10th International Workshop on Computational Systems Biology (WCSB) 2013: Bioinformatics. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/14/S10

Authors' details
[1]Department of Signal Processing, Tampere University of Technology, 33720 Tampere, Finland. [2]Biozentrum, Universität Basel, 4056 Basel, Switzerland.

Published: 12 August 2013

References

1. Mocellin S, Provenzano M: **RNA interference: learning gene knock-down from cell physiology.** *Journal of Translational Medicine* 2004, **2.**
2. Agaisse H, Burrack L, Philips J, Rubin E, Perrimon N, Higgins D: **Genome-wide RNAi screen for host factors required for intracellular bacterial infection.** *Science* 2005, **309(5738)**:1248-1251.
3. Wollman R, Stuurman N: **High throughput microscopy: from raw images to discoveries.** *Journal of Cell Science* 2007, **120(21)**:3715-3722.
4. Yan P, Zhou X, Shah M, Wong S: **Automatic segmentation of high-throughput RNAi fluorescent cellular images.** *IEEE Transactions on Information Technology in Biomedicine* 2008, **12**:109-117.
5. Chen C, Li H, Zhou X, Wong S: **Constraint factor graph cut based active contour method for automated cellular image segmentation in RNAi screening.** *Microscopy* 2008, **230(2)**:177-191.
6. Farhan M, Ruusuvuori P, Emmenlauer M, Rämö P, Dehio C, Yli-Harja O: **Graph cut and image intensity-based splitting improves nuclei segmentation in high-content screening.** *Proc SPIE 8655, Image Processing: Algorithms and Systems XI* 2013.
7. Wählby C, Lindblad J, Vondrus M, Bengtsson E, Björkesten L: **Algorithms for cytoplasm segmentation of fluorescence labelled cells.** *Analytical Cellular Pathology* 2002, **24(2-3)**:101-111.
8. Held C, Palmisano R, Häberley L, Hensel M, Wittenberg T: **Comparison of parameter-adapted segmentation methods for fluorescence micrographs.** *Cytometry, Part A* 2011, **79(11)**:933-945.
9. Lindblad J, Wählby C, Bengtsson E, Zaltsman A: **Image analysis for automatic segmentation of cytoplasms and classification of Rac1 activation.** *Cytometry, Part A* 2004, **57**:22-33.
10. Garrido A, de la Blanca NP: **Applying deformable templates for cell image segmentation.** *Pattern Recognition* 2000, **33(5)**:821-832.
11. Brox T, Weickert J: **Level set based image segmentation with multiple regions.** *Pattern Recognition, Springer LNCS* 2004, **3175**:415-423.
12. Vese L, Chan T: **A multiphase level set framework for image segmentation using the Mumford and Shah model.** *Computer Vision* 2002, **50(3)**:271-293.
13. Allalou A, van de Rijke F, Tafrechi R, Raap A, Wählby C: **Image based measurements of single cell mtDNA mutation load.** *Springer LNCS* 2007, **4522**:631-640.
14. Leskó M, Kato Z, Nagi A, Gombos I, Török Z, Vigh L, Vigh L: **Live cell segmentation in fluorescence microscopy via graph cut.** *Proc IEEE International Conference on Pattern Recognition* 2010, 1485-1488.
15. Russel C, Metaxas D, Restif C, Torr P: **Using the P^n potts model with learning methods to segment live cell images.** *Proc IEEE International Conference on Computer Vision* 2007, 1-8.
16. Quelhas P, Marcuzzo M, Mendonça AM, Campilho A: **Cell nuclei and cytoplasm joint segmentation using the sliding band filter.** *IEEE Transactions on Medical Imaging* 2010, **29(8)**:1463-1473.
17. Brejl M, Sonka M: **Edge based image segmentation: machine learning from examples.** *Proc IEEE International conference on Neural Networks. IEEE World congress on computational intelligence* 1998, 814-819.
18. Prasad M, Zisserman A, Fitzgibbon A, Kumar MP, Torr PHS: **Learning class-specific edges for object detection and segmentation.** *Proc Indian conference on Computer Vision, Graphics and Image Processing* 2006, 94-105.
19. Keuper M, Bensch R, Voigt K, Dovzhenko A, Palme K, Burkhardt H, Ronneberger O: **Semi-supervised learning of edge filters for volumetric image segmentation.** *DAGM-Symposium* 2010, 462-471.
20. Brejl M, Sonka M: **Object localization and border detection criteria design in edge-based image segmentation: automated learning from examples.** *IEEE Transactions on Medical Imaging* 2000, **19(10)**:973-985.
21. Zuiderveld K: *Contrast limited adaptive histogram equalization* Graphic Gems IV San Diego: Academic Press Professional; 1994.
22. Russ JC: *The image processing handbook.* 3 edition. CRC Press; 1999.
23. Selinummi J, Ruusuvuori P, Podolsky I, Ozinsky A, Gold E, Yli-Harja O, Aderem A, Shmulevich I: **Bright field microscopy as an alternative to whole cell fluorescence in automated analysis of macrophage images.** *PLoS One* 2009, **4(10)**:e7497.
24. Lindeberg T: **Scale-space theory: a basic tool for analysing structures at different scales.** *Journal of Applied Statistics, Supplement Advances in Applied Statistics: Statistics and Images: 2* 1994, **21(2)**:225-270.
25. Otsu N: **A threshold selection method from gray-level histograms.** *IEEE Transactions on Systems, Man and Cybernetics* 1979, **9**:62-66.
26. Ruusuvuori P, Manninen T, Huttunen H: **Image segmentation using sparse logistic regression with spatial prior.** *Proc IEEE European Signal Processing Conference* 2012, 2253-2257.
27. Ojala T, Pietikäinen M, Mäenpää T: **Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.** *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2002, **24(7)**:971-987.
28. Jain AK, Farrokhnia F: **Unsupervised texture segmentation using Gabor filters.** *Proc IEEE International conference on Systems, Man and Cybernetics* 1990, 14-19.
29. Haralick RM, Shanmugam K, Dinstein I: **Textural features for image classification.** *IEEE Transactions on Systems, Man and Cybernetics* 1973, **3(6)**:610-621.
30. Tibshirani R: **Regression shrinkage and selection via the Lasso.** *Journal of the Royal Statistical Society, Series B* 1994, **58**:267-288.
31. Friedman JH, Hastie T, Tibshirani R: **Regularization paths for generalized linear models via coordinate descent.** *Journal of Statistical Software* 2010, **33**:1-22.
32. Carpenter AE, Jones TR, Lamprecht MR, Clarke C, Kang IH, Friman O, Guertin DA, Chang JH, Lindquist RA, Moffat J, Golland P, Sabatini DM: **CellProfiler: image analysis software for identifying and quantifying cell phenotypes.** *Genome Biology* 2006, **7(10)**:R100.
33. **Broad Bioimage Benchmark Collection Website.** 2013 [http://www.broadinstitute.org/bbbc/BBBC007/].
34. Xiong G, Zhou X, Ji L, Bradley P, Perrimon N, Wong S: **Segmentation of drosophila RNAi fluorescence images using level sets.** *Proc IEEE International Conference on Image Processing* 2006, 73-76.
35. Jones TR, Carpenter A, Golland P: **Voronoi-based segmentation of cells on image manifolds.** *ICCV Workshop on Computer Vision for Biomedical Image Applications* 2005, 535-543.
36. Farhan M, Yli-Harja O, Niemistö A: **A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search.** *Pattern Recognition* 2013, **46**:741-751.
37. Danek O, Matula P, de Solorzano CO, Munoz-Barrutia A, Maska M, Kozubek M: **Segmentation of touching cell nuclei using a two-stage graph cut model.** *Springer LNCS* 2009, **5575**:410-419.

doi:10.1186/1471-2105-14-S10-S6
Cite this article as: Farhan *et al.*: **Multi-scale Gaussian representation and outline-learning based cell image segmentation.** *BMC Bioinformatics* 2013 **14**(Suppl 10):S6.

Publication II

M. Farhan, P. Ruusuvuori, M. Emmenlauer, P. Rämö, O. Yli-Harja, and C. Dehio, "Graph cut and image intensity-based splitting improves nuclei segmentation in high-content screening," in *Proceedings of SPIE 8655, Image Processing: Algorithms and Systems XI, 86550F*, San Francisco, USA, February 3-7, 2013, 10p.

Graph cut and image intensity-based splitting improves nuclei segmentation in high-content screening

Muhammad Farhan[a], Pekka Ruusuvuori[a], Mario Emmenlauer[b], Pauli Rämö[b], Olli Yli-Harja[a], Christoph Dehio[b]

[a]Department of Signal Processing, Tampere University of Technology, Tampere, Finland;
[b]Biozentrum, Universität Basel, Basel, Switzerland.

ABSTRACT

Quantification of phenotypes in high-content screening experiments depends on the accuracy of single cell analysis. In such analysis workflows, cell nuclei segmentation is typically the first step and is followed by cell body segmentation, feature extraction, and subsequent data analysis workflows. Therefore, it is of utmost importance that the first steps of high-content analysis are done accurately in order to guarantee correctness of the final analysis results. In this paper, we present a novel cell nuclei image segmentation framework which exploits robustness of graph cut to obtain initial segmentation for image intensity-based clump splitting method to deliver the accurate overall segmentation. By using quantitative benchmarks and qualitative comparison with real images from high-content screening experiments with complicated multinucleate cells, we show that our method outperforms other state-of-the-art nuclei segmentation methods. Moreover, we provide a modular and easy-to-use implementation of the method for a widely used platform.

Keywords: High-content screening, Image segmentation, Graph cut, Concavity point analysis, Clump splitting.

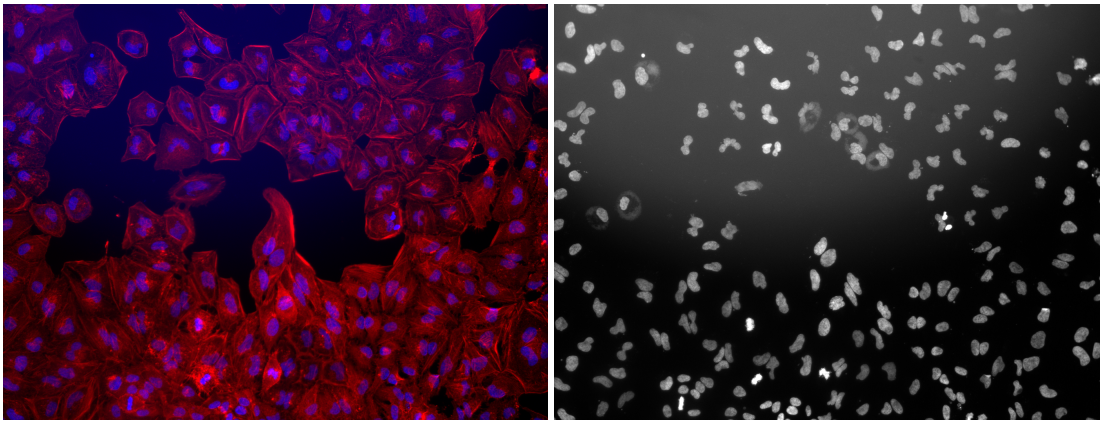
1. INTRODUCTION

High-content screening for drug discovery demands a huge amount of image data to be captured and analyzed automatically in a batch process. Segmentation of those images is typically the first step in high-content image analysis. If this is not optimally performed, all the following analysis steps may yield inaccurate results. Segmentation of cell microscopy images not only comprises separation of cell body from the background but also includes separation of cells from each other. Although, the former step seems trivial, yet general image segmentation methods are not always applicable to most of the cases. The later step, however, is non-trivial and is called clump splitting which is required since groups of two or more cells detected as a single object (called as clumps later in the article) often persist after the initial segmentation step. In order to study and quantify the viral entry into a cell, single cell analysis is needed. Clump splitting is crucial for performing single cell analysis.

Many of the methods for high-content image analysis, for example,[1,2] perform cell image segmentation in two steps: nuclei segmentation and cytoplasm segmentation, where the result of the former step is used to perform the latter step. However, these methods focus mainly on cytoplasm segmentation even though errors in nuclei segmentation are even more problematic. Moreover, the accuracy of cell body segmentation is dependent on the accuracy of nuclei segmentation step. For example, the clumps are confronted with not only at the cytoplasm level but also at the nuclei level and better cytoplasm separation is only ensured by robust nuclei clump splitting. Figure 1(a) shows an example where the cell boundaries are so indiscernible that nuclei channel image is needed to detect them. However, from Figure 1(b) it is clear that sometimes even nuclei are hard to separate, setting a challenge for accurate cell segmentation.

The nuclei image segmentation methods found in the literature for fluorescent microscopy images include classic segmentation methods and active contour methods. Methods based on, or a combination of, Otsu thresholding, seeded watershed algorithm with or without h-maxima transform[1–4] and morphological filtering incorporating gradient and shape information[5,6] are widely used. On the other hand, Graph cut methods incorporating image gradient and shape information[7] and level set based algorithms[8,9] incorporating Radon transform for cell nuclei separation[10] have also been used.

Further author information: (Send correspondence to M.F.) M.F.: muhammad.farhan@tut.fi, P.R.: pekka.ruusuvuori@tut.fi, M.E.: mario.emmenlauer@unibas.ch, P.R.: pauli.ramo@unibas.ch, O.Y-H.: olli.yli-harja@tut.fi, C.D.: christoph.dehio@unibas.ch



(a) *Nuclei and Cytoplasm Image* (b) *Nuclei Image*

Figure 1: (a) A cell microscopy image showing nuclei in Blue and cell body in Red. (b) The nuclei channel image, a representative of the problematic cases and highlighting the problems confronted in their segmentation such as uneven illumination, noise around nuclei contour, out of focus nuclei, clumps of nuclei etc. The size of the image is 1040 x 1392 pixels.

The existing nuclei image segmentation methods, when applied individually to our high-throughput fluorescent microscopy images, are sometimes observed to produce fusion as well as cutting of nuclei along with suboptimal separation of touching nuclei. Also they fail to detect and restore multiple nuclei cells and fuse butterfly-shaped nuclei belonging to some cell phenotypes. This results in loss of many interesting biological phenotypes. Inability to find all the phenotypes causes their misclassification, which leads to inaccurate subsequent biological analysis. The problems get further aggravated when the imaging condition causes uneven illumination as well as some cells to be out of focus. Figure 1 shows an image from the data set in which most of the aforementioned problems are noticeable.

We observed that if satisfactory foreground/background segmentation is obtained using the existing segmentation methods, that is, restricting them from over-splitting by not trying to resolve clumps, then a robust clump splitting method can produce a better overall segmentation result. Therefore, in order to solve most of the aforementioned problems, we present a novel framework for cell nuclei image segmentation which utilizes robust graph cut-based method to get foreground/background segmentation and our image intensity-based clump splitting method to deal with under-segmentation.

The rest of the paper is organized as follows: Section 2 describes the proposed cell nuclei image segmentation framework. The details of experimentation, results and discussions are presented in Section 3. The implementation of the method is discussed in Section 4. Section 5 concludes the paper.

2. NUCLEI IMAGE SEGMENTATION FRAMEWORK

The problems associated with high-throughput fluorescent imaging systems is that quite often the images have uneven illumination, out of focus cells as well as part of the image where the contrast is very low. Graph cut method,[7,11,12] described below, is robust enough to tackle most of these issues, however, contrast enhancement is very much needed as few nuclei are dark enough that their intensity values almost touch the background image intensity level. Here, we performed contrast enhancement using adaptive histogram equalization followed by morphological erosion based reconstruction.

2.1 Foreground/Background Segmentation using Graph Cut

Here we describe the graph cut-based energy minimization technique for finding the optimal foreground/background image segmentation. Image segmentation can be considered as a task of labeling the individual pixel $p \in P$ of the image having intensity I_p with the label $L_p \in \{0, 1, 2, ...M\}$ based on certain constraints.[11,12] Label assignment can be constrained by defining it in the form of an energy function which comprises the cost of assigning wrong labels to individual pixels as well as the cost of assigning different labels to neighboring pixels. A set of labels which yields a minimum of that function or, in other words, minimizes the cost of labeling gives the optimal solution. One of such energy functions is given by the Potts model as,[12]

$$E(L) = \sum_{p \in P} D_p\left(L_p\right) + \sum_{(p,q) \in N} V_{p,q} \cdot \delta_{(L_p \neq L_q)}, \tag{1}$$

where $D_p(L_p)$ represents the cost of assigning the label L_p to the pixel p, $(p,q) \in N$ is the neighborhood system, and $V_{p,q}$ represents the cost of discontinuity occurring due to assignment of different labels to the neighboring pixels.[12] The second term is based on Markov Random Fields(MRF) assumption with additional constraint that weights or cost of only differing neighbors are considered.

Maximum a Posteriori (MAP) estimate of a labeling with MRF assumption governs that $D_p(L_p)$ can be modeled as a likelihood function[11] given by

$$D_p(L_p) = -\ln \Pr(I_p|L_p), \tag{2}$$

where $\Pr(I_p|L_p)$ specifies the probability of a pixel with intensity I_p when it belongs to label L_p and is assumed to be known *a priori*. The peaks of the foreground and background pixels in the image intensity histogram can be used to set the preference of assigning a label (binary) to a pixel. That is, the probability that a pixel with particular intensity value is either foreground or background pixel can be deduced by hard-thresholding the image intensity histogram. Therefore, on the basis of that probability, the value of $D_p(L_p)$ is set either 0 or K(instead of $+\infty$, due to implementation point of view).

Since $V_{p,q}$ penalizes for the discontinuity arising due to assigning different labels to neighboring pixels p and q, therefore, it can be obtained using intensity gradient. As a matter of fact, the penalization should be higher or, in other words, the cost should be bigger if pixels p and q with similar intensity value are assigned different labels. Moreover, distance between pixels must also be incorporated since the cost should decrease when the pixels are distant. Boykov *et al.* in[11] takes these dependencies into account to give the expression for $V_{p,q}$ as

$$V_{p,q} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}, \tag{3}$$

where $dist(.)$ is the distance function and σ is the estimated average gradient magnitude in the image and controls the penalization. Euclidean metric can be used but Riemannian metric produces geodesic contour with minimal artifacts.[7,13]

To obtain the foreground/background segmentation, exact minimization of the energy function in (1) is required. Greig *et al* in[14] showed that it can be achieved by finding a minimum-cost cut in a two-terminal directed graph, see for example in Boykov *et al.*[12] In such a setting, each image pixel in a neighborhood N is denoted by a node which is connected with the other nodes through edges called n-links. Their edge weights are defined by $V_{p,q}$ because of its penalization effect on the discontinuity between pixels. For the two labels, two terminal nodes called source s and sink t are formed and all the pixel nodes are connected with them through edges called t-links. Their edge weights are defined by $D_p(L_p)$ because of its penalization effect on assigning a label to a particular pixel.

Next, an optimal source/sink cut C is desired which separates the graph nodes into two disjoint sets S and T with s and t respectively such that the sum of the costs is minimal. Since the cost of (p,q) and (q,p) may differ, therefore, to find an optimal cut with minimal cost, all the cutting possibilities need to be considered.[12] This can be formulated as a problem of finding the maximum flow of water from the source to sink through the edges, imagining them as pipes and their weights as the capacity with which the water can pass through. A solution to this problem can be found in polynomial time using a maximum flow algorithm[12] which gives us the desired foreground/background segmentation. Here we have used the implementation presented in Daněk *et al.*[7]

2.2 Image Intensity-based Clump Splitting

In order to perform single cell analysis, resolution of clumps of cells into constituent single cells is necessary. In Farhan *et al.*[15] we presented a novel nonparametric concavity point analysis-based clump splitting method for convex objects. The method not only incorporates the holes present in the clumps but also the image intensity, if it has enough variation along the region where the objects clump together, for finding the split lines. However, the results of this approach depend on the accuracy of the initial image segmentation. Hence it was necessary to get the initial segmentation performed using a graph cut-based method which is known to be robust. The clump splitting method itself does not need parameters, but we have applied a size limit in order to constraint the minimum size for the objects considered for splitting. Below we describe the steps performed by the clump splitting method of.[15]

Concavity and Hole Prominent Point Detection

In a concavity point analysis-based clump splitting method, the first and the most important step is the accurate detection of all the concavity points (red squares in Figure 2). They are the points on the clumped object contour where the contour

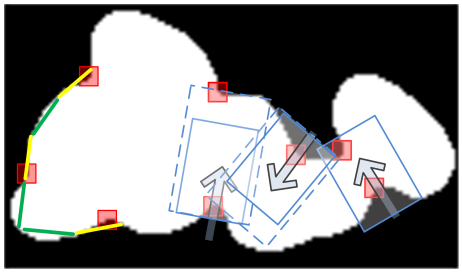


Figure 2: Concavity point detection on the left where contour segment traced by green and yellow lines are convex and non-convex respectively. Concavity point-pair search on the right where directional vector and corresponding window mask is used for finding pair of a concavity point.

of two objects meet together. They are obtained from those contour segments which cease to be convex, see for example, segments corresponding to yellow lines in Figure 2. Such contour segments are found by visualizing a straight line between two contour points, 20 contour pixels apart*, starting from the top-leftmost contour pixel in clockwise direction. If the line passes through the background, that particular contour segment is considered non-convex and a new line is visualized between end points of next such contour segment beginning from the fourth contour point in the previous segment. In case of the line residing inside the object, that contour segment is considered convex and a new line is visualized between end points of next such contour segment beginning from the end of the previous contour segment, until the starting point is reached again. For the non-convex contour segments, the point along the contour segment with the largest distance from its local chord is picked provided that distance is greater than 2 pixels (always constant) and also the mid point of the chord lies on the background. These conditions ensure detection of only valid concavity points and reject points resulting due to boundary irregularities. The local chord for a particular point is obtained by joining 6^{th} contour points on either side of the point.

When more than two objects clump together in a complicated way, they tend to create holes within the clumps. Detection of such holes and the prominent points on their contour is also necessary in the initial phase for accurate resolution of complex clumps. This is because those points can be joined with another such point or with a concavity point to get the split line. Those prominent points are also found in the similar way to the concavity points.

Concavity Point-Pair Search

Once all the concavity points are found, the next step is to find the appropriate point-pairs which could be joined to split the clumps. The pairing point to a concavity point need not be another concavity point rather it can be a prominent point on the contour of a hole or a point on the clumped object contour opposite to the concavity point. In order to avoid confusion, from now on the term concavity point will be used for hole prominent points too. Using feature or rule-based approach for finding the pairs of concavity points often leads to some concavity points without a pairing point. Rather, the greedy approach of looking for the best pair for each concavity point works quite well.[15]

One of the features associated with a concavity point is the directional vector which is a vector with its head on the concavity point and tail on the mid-point of its local chord, such that the vector bisects the concavity region around the concavity point, see for example, blue arrows in Figure 2. It was observed that the pairing point is usually found in a particular area in the direction of the directional vector. This amounts to a variable size rectangular window (blue rectangles in Figure 2) oriented in the direction of the directional vector. Such window is constructed to be used as a mask for finding the pairing point of a split line. The window height or length is kept to a certain value based on the size of the longest split line that is allowed in an image set. This comes from the constraint for the smallest allowed object in the resulting images or the minimum size of the object considered for clump splitting. The width of the window is kept to a small value initially, only to be varied until a pair is found. The small window width helps to avoid the situation of having two pairing points in the window, a case, in which the point less distant to the concavity point under observation is retained. This approach of search window makes the method independent of the user-defined parameters as well as user-defined threshold values for features to find split lines even if the image set is large and contains objects of varying sizes, shapes and complexity of clumps.

*This value of 20 pixels is chosen empirically from very large data sets with real and synthetic microscopy images having convex objects of varying size, shape and clumped together with varying amount and probability of overlap.[15]

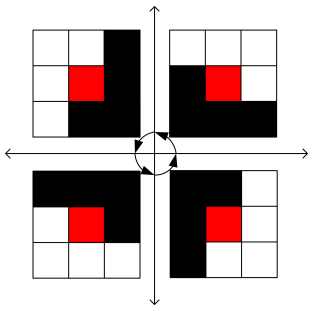


Figure 3: Four of the possible binary masks, based on the quadrant in which angle of the directional vector lies, to search for minimum/maximum split path. The black and red pixels are Don't Care.

Split Line Formation

If there is intensity variation in the region where objects clump then it can be used to find the split lines. However, when the image intensity is not used or is deemed ineffective for clump splitting, the point-pairs found in the previous step are joined together to get the splitting done. It has been observed that biological microscopy images often have intensity variations that can be effectively used as extra information for clump splitting. Moreover, since the usage of image intensity in finding the split path for clump splitting provides much improved results, especially for more complex clumps as well as for clumps near the image borders,[15] therefore, it is the essence of our nuclei image segmentation framework. In this approach, a split line is obtained for every concavity point by using its directional vector. It guides the search for the path of minimum/maximum intensity between the concavity point and a point on another split line or on object contour in another concavity region.[15] Starting from the concavity point, a particular mask or kernel is used to find the next lowest/highest intensity pixel in the 3x3 neighborhood of the current pixel. This resembles sliding window-based search but differs in the sense that it is directional search and not all the pixels are operated upon. Based on the angle of directional vector with respect to x-axis, one of the four different 3x3 kernels of Figure 3 is used. For every concavity point, the search continues until another concavity point or a background pixel is reached. The advantage of this approach is that it can accurately split complex clumps with true object areas similar to the one obtained by a human observer.

3. EXPERIMENTAL RESULTS AND DISCUSSION

All the data used in this paper come from the genome-wide high-content screens performed under InfectX[†]. The biological and technical details of the screens will be published separately. The original image set comprises a huge amount of DNA-channel nuclei images. However, for quantitative evaluation of our method and for its comparison with other methods, corresponding ground truth information is needed which is not available. Therefore, due to the necessity of creating ground truth labeling manually, we have to restrict our test set to few images but enough to cover all the problematic cases. Here we used 10 most problematic images identified by the biologists based on the currently available segmentation results from CellProfiler.[16] The images contained $\sim$2000 nuclei appearing as either separate nucleus or as a clump of nuclei with the probability of overlap value of 0.20. Once the ground truth labeling is performed and authenticated by the biologists, segmentation methods are applied to the images and the obtained results are compared with the ground truth to obtain quantitative measures. The segmentation accuracy is measured by F-measure (FM) which is the harmonic mean of Precision (PR) and Recall (RC) and is given by

$$\mathrm{FM} = \frac{2}{\left(\frac{1}{\mathrm{PR}} + \frac{1}{\mathrm{RC}}\right)}, \quad \text{where } \mathrm{PR} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FP})} \text{ and } \mathrm{RC} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})}, \tag{4}$$

where TP, FP and FN are true positive (object was in the ground truth and detected), false positive (object was not in the ground truth but detected) and false negative (object was in the ground truth but not detected) respectively. Higher segmentation accuracy is achieved by lower values of FP and FN.

[†]InfectX is a consortium of 11 research groups, covering bacterial entry, viral entry, proteomics and modeling. The goal of InfectX is to comprehensively identify the components of the human infectome for a set of important bacterial and viral pathogens and to develop new mathematical and computational methods with predictive power to reconstruct key signaling pathways controlling pathogen entry into human cells. To date, InfectX has performed several genome-wide high-content siRNA screens for several pathogens. A current focus point is to perform high-quality image analysis including state-of-the-art object segmentation.

Table 1: Performance parameters for four methods before and after clump splitting. For example, TP1, FP1, ..., FM1 are values before applying clump splitting whereas TP2, FP2, ..., FM2 etc. are the values after clump splitting. See the text for more details about the abbreviations.

Performance Parameters												
-	TP1	FP1	FN1	PR1	RC1	FM1	TP2	FP2	FN2	PR2	RC2	FM2
GC	1559	8	420	0.995	0.788	0.879	1934	85	45	0.958	0.977	0.967
LS	1617	19	362	0.988	0.817	0.895	1943	121	36	0.941	0.982	0.961
MG	1752	55	227	0.970	0.885	0.925	1844	217	135	0.895	0.932	0.913
CS	1876	221	103	0.895	0.948	0.920	-	-	-	-	-	-

In order to quantitatively evaluate our method, first we obtained the results by segmenting the images using graph cut method (GC) along with other state-of-the-art image segmentation methods based on level set (LS),[8] image gradient and morphological filtering (MG),[5] and classical segmentation (CS) method from CellProfiler.[16] Clump splitting is built-in in CS using morphological watershed as well as in MG using gradient and negative Laplacian of Gaussian. The performance parameters for this initial segmentation are given in Table 1 (Left side). As the idea is to get robust initial segmentation so GC and LS are bound to perform under-splitting which is evident from their higher FN and lower FP values. The major contribution in the higher FN values is from the nuclei that are clumped together and detected as a single object rather than being detected as separate objects, which can later be tackled by robust clump splitting. On the other hand, in spite of embedded clump splitting in MG and CS, their FN values are quite higher which means they are still not good enough to detect the individual nuclei from clumps. Moreover, higher FP values for MG and CS indicate that in the quest of performing clump splitting they sometimes perform over-splitting.

Next, we applied clump splitting to all the methods except CS. The need of applying clump splitting to MG arose due to the fact that there are few unresolved clumps from it. The performance parameters for this step are given in Table 1 (Right side). It is evident from the table that once clump splitting is applied to GC and LS their overall results improve significantly. A substantial decrease in the values of FN for GC and LS means that the clump splitting method is quite robust in splitting the nuclei clumps. However, a slight increase in the values of FP for the two methods indicates the inherent trade-off between under- and over-segmentation. On the other hand, value of FN for MG decreased a lot but also at the expense of FP value. One of the other reasons for higher FP value here is the inaccurate initial segmentation, especially in cases of noise around nuclei. That causes a lot of noisy structures in the initially segmented results which gives false objects when passed through clump splitting phase, see, for example, Figure 5 and Figure 6.

Based on the quantitative measures from Table 1 (see also Figure 4), GC-based and LS-based methods clearly outperform the other methods with the proposed (GC + Clump Splitting) giving an FM value of $\sim$0.97 with (LS + Clump Splitting) comparable to it. Although (LS + Clump Splitting) gives FM value similar to the proposed (GC + Clump Splitting) but it gives much more over-splitting as evident from its higher FP value, FP2, in Table 1 and also from Figure 7. Not only in quantitative measures, the overall segmentation from the proposed (GC + Clump Splitting) is much better than others in qualitative measures[‡] as well. From the example images it seems that the initial segmentation results of all the methods are similar, however, it actually is not the case. Magnifying the images a little reveals that GC provides the smoothest contour which is a requirement for the clump splitting method to produce accurate final segmentation results. This is also the reason why clump splitting gives more accurate results, especially qualitative results, for GC than for others. On the other hand, both LS and MG methods tend to give noisy initial segmentation especially in case of protrusion nuclei or a cell which is highly infected, see, for example, Figure 5 (nuclei along main diagonal) and Figure 6 (nuclei on top-left and bottom-right). CS gives good segmentation occasionally and produces enough over- and under-splitting. Figure 7 compares the results from all the methods with and without clump splitting where it is obvious that the proposed (GC + Clump Splitting) is giving very few erroneously detected nuclei compared to other methods.

Further insight into Table 1 (Right side) leads to the point that although the FM values suggest that the proposed (GC + Clump Splitting) is almost similar to (LS + Clump Splitting) and gives $\sim 5\%$ increase in accuracy as compared to MG and CS based methods, but for a total number of $\sim$2000 cell nuclei, (LS + Clump Splitting), (MG + Clump Splitting) and CS are giving around 30, 220 and 190 more falsely detected nuclei (FP+FN value), respectively, in comparison with (GC + Clump Splitting). When it comes to overall cell segmentation even this much falsely detected nuclei can make a big

[‡]Due to the absence of ground truth overall segmentation results, the qualitative assessment is based on the accuracy of the overall segmentation results against susceptibility to noise and is performed manually by the experts.

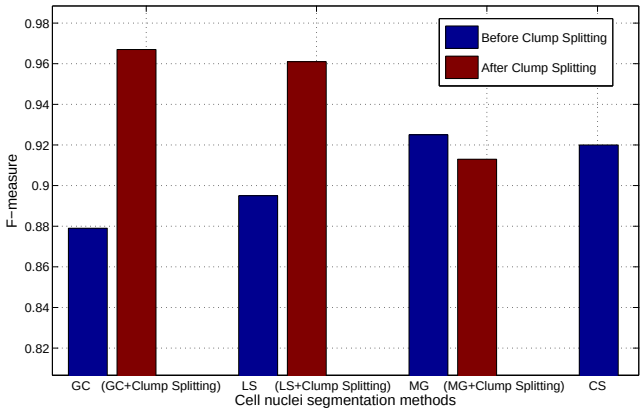
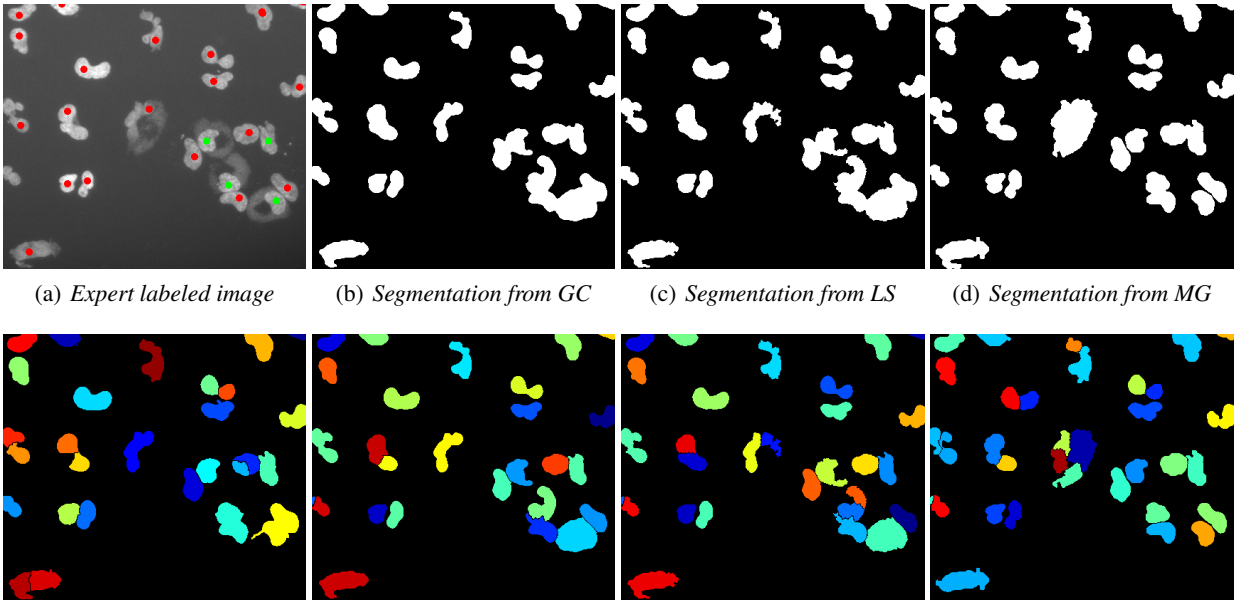


Figure 4: F-measure (FM) for Graph cut (GC), Level set (LS), Gradient-based (MG) and CellProfiler-based (CS) methods before and after application of clump splitting.

difference. This is another evidence of the authoritative performance of our proposed framework of (GC+ClumpSplitting). It shows that the proposed framework is able to produce the desired results which any of the other methods compared here, especially the method previously in use, is unable to produce.

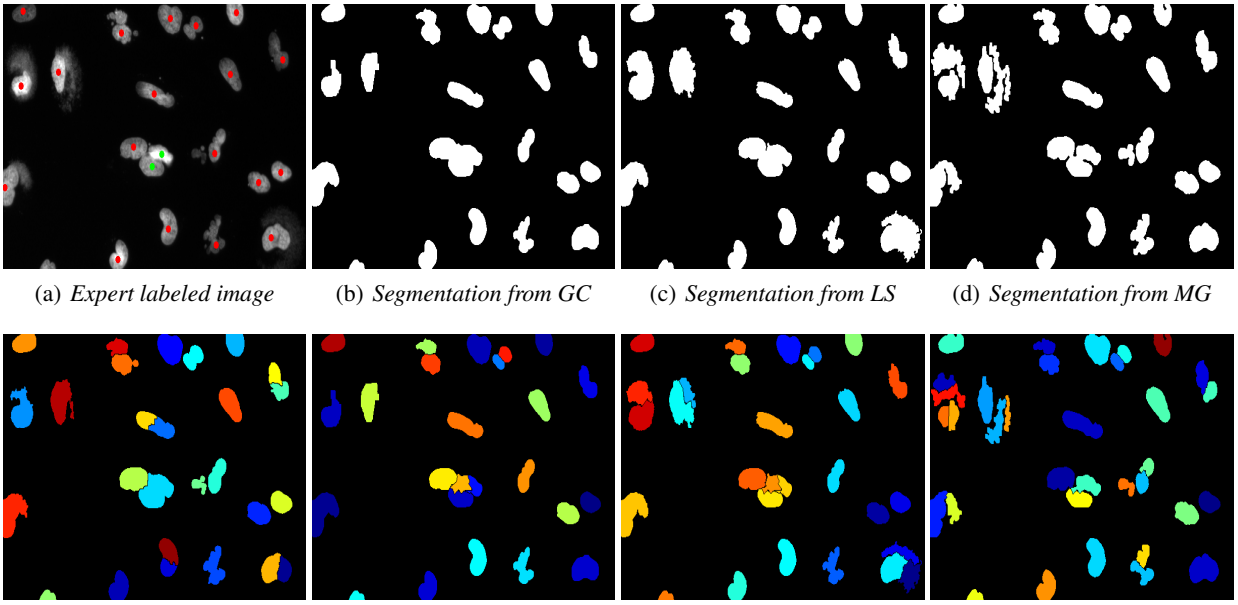
Finally, it is worth-mentioning that the labeling was done in such a way that nuclei belonging to multinuclear cells are treated as one object. However, our robust clump splitting method tends to split these objects. This can be incorporated in the future work where segmenting the cell body a feedback system is developed to improve nuclei segmentation by indicating the nuclei which need to be merged.



(a) *Expert labeled image* (b) *Segmentation from GC* (c) *Segmentation from LS* (d) *Segmentation from MG*

(e) *Result from CS* (f) *(GC+ClumpSplitting)* (g) *(LS+ClumpSplitting)* (h) *(MG+ClumpSplitting)*

Figure 5: Qualitative comparison of cell nuclei image segmentation methods. In (a), a red dot means a separate nucleus and a green dot along with red dot means clump of nuclei. (b)-(d) Segmentation results from GC, LS and MG without clump splitting. (e)-(h) Segmentation results from CS, proposed (GC + Clump Splitting), (LS + Clump Splitting) and (MG + Clump Splitting).



(a) *Expert labeled image* (b) *Segmentation from GC* (c) *Segmentation from LS* (d) *Segmentation from MG*

(e) *Result from CS* (f) *(GC+ClumpSplitting)* (g) *(LS+ClumpSplitting)* (h) *(MG+ClumpSplitting)*

Figure 6: Qualitative comparison of cell nuclei image segmentation methods. In (a), a red dot means a separate nucleus and a green dot along with red dot means clump of nuclei. (b)-(d) Segmentation results from GC, LS and MG without clump splitting. (e)-(h) Segmentation results from CS, proposed (GC + Clump Splitting), (LS + Clump Splitting) and (MG + Clump Splitting).

4. MODULE IMPLEMENTATION IN CELLPROFILER 1.0

The objective of this study was not only to develop a method to improve cell nuclei segmentation but also to emphasize its usage in high-content screening experiments. Such settings requires a full pipeline of image analysis methods accompanied by subsequent data analysis tools. Thus, it is reasonable to use an open source software tool which has a modular structure, enabling to expand the existing library of analysis tools and to replace parts of the standard analysis pipeline with new tools. The tendency towards user and developer-friendly, modular open-source software tools enables constant development by the community.[17] Here we choose CellProfiler 1.0[16] as the platform due to its wide usage in high-content screening studies and its easy extendibility.[18–20] A CellProfiler module is developed which can be used in an image analysis pipeline, where the module replaces nuclei segmentation module of CellProfiler, for high-throughput analysis. This helps in validation of the proposed cell nuclei segmentation method as well as to show its promising performance for a very large image set. In addition to that, with the provision of using different methods for segmentation, by including them in the module, it also helps in comparing the results of our proposed method. Nuclei segmentation is only a part of our goal of building an efficient image analysis pipeline for automated and quantitative analysis of high-content screens. Thus, as we get more results from experiments, the implementation is likely to be improved based on the gained experience. The current version of the implementation, however, is available on request. The module is implemented in Matlab except the graph cut part which is in C++.[7] On a 3.0 GHz CPU with 4GB RAM, 64Bit Win 7, the whole nuclei segmentation process takes on average 5 seconds per image for a dataset containing images with around 200 nuclei with varying probabilities of overlap.

5. CONCLUSION

In high-content screening experiments, quantifying the amount of viral entry to a cell requires accurate cell segmentation. In a prelude to whole cell segmentation, we proposed a cell nuclei segmentation framework where graph cut separates foreground from background and a subsequent step of clump splitting segregates cell nuclei from their clumps. Comparison of our method against a set of state-of-the-art nuclei segmentation methods reveal that it outperforms them not only in terms of quantitative measures but qualitatively as well. A module of the method is developed in an open platform and it is tested with image analysis pipeline to highlight the promising performance of our method and to make it available for routine use in microscopy image analysis. The future work contains using the result of nuclei segmentation as context for the cytoplasm segmentation and also propagating the method for its usage in routine high-content screening analysis.

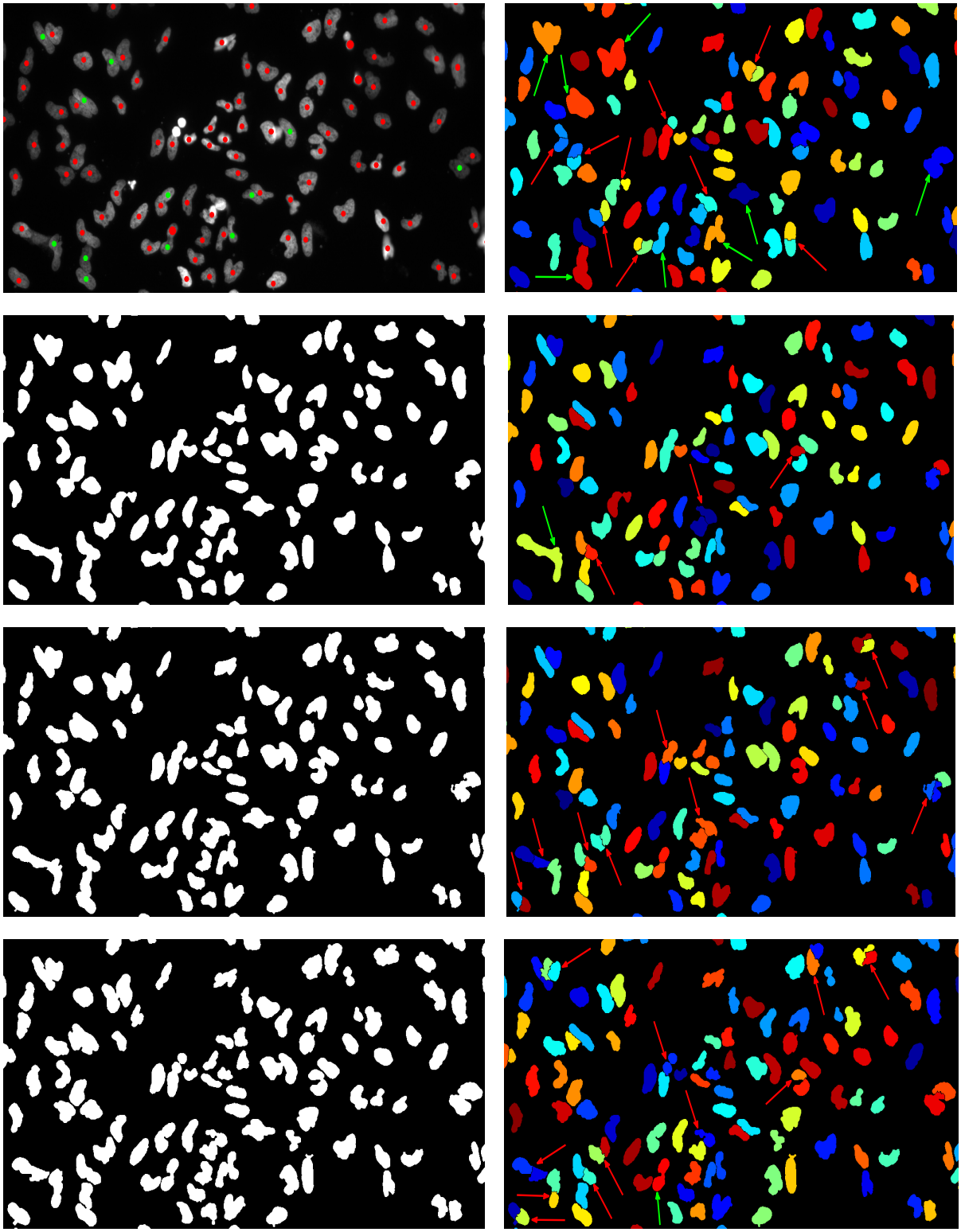


Figure 7: Qualitative comparison of cell nuclei image segmentation methods. In (top left), a red dot means a separate nucleus and a green dot along with red dot means clump of nuclei. In left Column: (Top to Bottom) Expert labeled image, Segmentation results from GC, LS and MG without clump splitting. In Right Column: (Top to Bottom) Segmentation results from CS, proposed (GC + Clump Splitting), (LS + Clump Splitting) and (MG + Clump Splitting). Red and Green arrows indicate over- and under-segmentation respectively.

ACKNOWLEDGMENTS

We acknowledge the financial support from TISE graduate school (MF), Academy of Finland project #140052 (PR), and grant 51RT-0-126008 (InfectX) in the frame of SystemsX.ch, the Swiss Initiative for Systems Biology (to Christoph Dehio). We are also very grateful to Shyan Huey Low and Alain Casanova for providing image unpublished data.

REFERENCES

[1] Chen, C., Li, H., Zhou, X., and Wong, S. T. C., "Constraint factor graph cut based active contour method for automated cellular image segmentation in RNAi screening," *Microscopy* **230**(2), 177–191 (2008).

[2] Yan, P., Zhou, X., Shah, M., and Wong, S. T. C., "Automatic segmentation of high-throughput RNAi fluorescent cellular images," *IEEE Transactions on IT in Biomedicine* **12**(1), 109–117 (2008).

[3] Lindblad, J., Wählby, C., Bengtsson, E., and Zaltsman, A., "Image analysis for automatic segmentation of cytoplasms and classification of Rac1 activation," *Cytometry* **57**(1), 22–33 (2004).

[4] Allalou, A., van de Rijke, F. M., Tafrechi, R. J., Raap, A. K., and Wählby, C., "Image based measurements of single cell mtDNA mutation load," *Springer LNCS* **4522**, 631–640 (2007).

[5] Matula, P., Kumar, A., Wörz, I., Erfle, H., Bartenschlager, R., Eils, R., and Rohr, K., "Single-cell-based image analysis of high-throughput cell array screens for quantification of viral infection," *Cytometry* **75**(4), 309–318 (2009).

[6] Wählby, C., Sintorn, I.-M., Erlandsson, P., Borgefors, G., and Bengtsson, E., "Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections," *Microscopy* **215**(1), 67–76 (2004).

[7] Daněk, O., Matula, P., de Solórzano, C. O., Muñoz-Barrutia, A., Maška, M., and Kozubek, M., "Segmentation of touching cell nuclei using a two-stage graph cut model," *Springer LNCS* **5575**, 410–419 (2009).

[8] Li, C., Xu, C., Gui, C., and Fox, M. D., "Distance regularized level set evolution and its application to image segmentation," *IEEE Transactions on Image Processing* **19**(12), 3243–3254 (2010).

[9] Maška, M., Daněk, O., de Solórzano, C. O., Muñoz-Barrutia, A., Kozubek, M., and Garcia, I. F., "A two-phase segmentation of cell nuclei using fast level set-like algorithms," *Springer LNCS* **5575**, 390–399 (2009).

[10] Dzyubachyk, O. M., Niessen, W. J., and Meijering, E., "Advanced level-set based multiple-cell segmentation and tracking in time-lapse fluorescence microscopy images," in [*IEEE International Symposium on Biomedical Imaging*], 185–188 (2008).

[11] Boykov, Y. and Funka-Lea, G., "Graph cuts and effcient N-D image segmentation," *International Journal of Computer Vision* **70**(2), 109–131 (2006).

[12] Boykov, Y. and Kolmogorov, V., "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(9), 1124–1137 (2004).

[13] Boykov, Y. and Kolmogorov, V., "Computing geodesics and minimal surfaces via graphcuts," in [*IEEE International Conference on Computer Vision*], 26–33 (2003).

[14] Greig, D., Porteous, B., and Seheult, A., "Exact maximum a posteriori estimation for binary images.," *Royal Statistical Society, Series B* **51**(2), 271–279 (1989).

[15] Farhan, M., Yli-Harja, O., and Niemistö, A., "A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search," *Pattern Recognition* **46**, 741–751 (2013).

[16] Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., Guertin, D. A., Chang, J. H., Lindquist, R. A., Moffat, J., Golland, P., and Sabatini, D. M., "Cellprofiler: image analysis software for identifying and quantifying cell phenotypes," *Genome Biology* **7**(10), R100 (2006).

[17] Carpenter, A. E., Kamentsky, L., and Eliceiri, K. W., "A call for bioimaging software usability.," *Nature Methods* **9**(7), 666–670 (2012).

[18] Snijder, B., Sacher, R., Rämö, P., Damm, E.-M., Liberali, P., and Pelkmans, L., "Population context determines cell-to-cell variability in endocytosis and virus infection.," *Nature* **461**, 520–523 (Sep 2009).

[19] Ruusuvuori, P., Aijö, T., Chowdhury, S., Garmendia-Torres, C., Selinummi, J., Birbaumer, M., Dudley, A. M., Pelkmans, L., and Yli-Harja, O., "Evaluation of methods for detection of fluorescence labeled subcellular objects in microscope images.," *BMC Bioinformatics* **11**, 248 (2010).

[20] Rämö, P., Sacher, R., Snijder, B., Begemann, B., and Pelkmans, L., "Cellclassifier: supervised learning of cellular phenotypes.," *Bioinformatics* **25**(22), 3028–30 (2009).

Publication III

M. Farhan, O. Yli-Harja, and A. Niemistö, "An improved clump splitting method for convex objects," in *Proceedings of the 7th International Workshop on Computational Systems Biology*, Luxembourg, June 16-18, 2010, pp. 35-38.

AN IMPROVED CLUMP SPLITTING METHOD FOR CONVEX OBJECTS

Muhammad Farhan, Olli Yli-Harja, and Antti Niemistö

Department of Signal Processing, Tampere University of Technology,
P.O. Box 553, FI-33101 Tampere, Finland
muhammad.farhan@tut.fi, olli.yli-harja@tut.fi, antti.niemisto@tut.fi

ABSTRACT

An improved version of the clump splitting method by Kumar *et al*. [4] is presented. The method is based on finding and linking concavity points to obtain lines that split concave clumps into their constituent convex objects. Images of yeast cells are used to compare the improved method quantitatively and qualitatively with the original method by Kumar *et al*. as well as with a widely used watershed-based method, and it is shown that the improved method removes the deficiencies present in the method by Kumar *et al.* and also performs better than the watershed method for clumps that have varying object sizes. Supplementary information can be found online at http://www.cs.tut.fi/sgn/csb/imclump/.

1. INTRODUCTION

Detecting individual objects from clumps of touching or overlapping objects by image analysis is important in many applications ranging from industrial conveyor belt automation [1] to detection of single biological cells from microscopic images [12]. Indeed, in many types of cell cultures the cells tend to form clusters or clumps. For example, yeast and many different bacteria typically grow in clumps.

Many segmentation methods are unable to resolve the individual cells from these clumps, and therefore methods for splitting clumps into their constituent cells are needed. This is typically done as a post-processing step after initial cell segmentation. The capability to accurately segment individual cells is needed in many kinds of single cell analyses. For example, the use of green fluorescent protein as a reporter of gene expression of single cells requires the ability to segment single cells [9].

There are many different methods for clump splitting. An important class of methods applies concavity analysis [3], [4], [10], [11]. These methods typically detect concavity points and link them to obtain lines that split object clumps into their constituent objects. Concavity points are points on the boundary of clustered image objects at which two convex objects come in contact with each other. Figure 1 shows concavity points (white) in a clump of four convex objects (black).

Our investigation of the methods found in the literature has revealed that they fail to split many clumps, especially the ones that occur in images of the budding yeast *Saccharomyces cerevisiae*. We studied the method from Kumar *et al.* [4] in which the main problem is that it does not consider the case of having more than one concavity point in a particular concavity region, which eventually leads to false split lines along with under-segmentation. Secondly there are problems in finding the candidate split lines because the measure of opposite alignment used by the method is not satisfactory for some of the more complex clumps. In this paper, we present an improved version of the clump splitting method by Kumar *et al*. where we removed the aforementioned deficiencies to make it applicable to more complex clumps.

The paper is organized as follows. Section 2 describes the improved clump splitting method. Section 3 presents the results and comparisons of the three different methods. Finally, Section 4 concludes the paper.

2. CLUMP SPLITTING METHOD

In this section, we describe the steps that are performed in the improved method to split the clumps. The deficiencies of the original method proposed by Kumar *et al*. [4] are highlighted, and the modifications that we made in order to remove those deficiencies are described. The method operates on binary images obtained from initial segmentation, and consists of three fundamental steps: detecting concavity points, listing candidate split lines, and finding the best split line for every concavity point.

2.1. Detecting concavity points

In every concavity point-based clump splitting method the first step, of course, is the detection of concavity points on the boundary of the objects. The method in [4] uses the approach in which all the concavity regions present in the object are first found. In Figure 1 such regions are marked as R_1-R_3, and the corresponding boundary segments and convex hull chords are represented by S_1-S_3 and K_1-K_3, respectively. The method then determines the concavity points by finding a point in every segment S_i which has the largest perpendicular distance from its corresponding K_i. This means that only one concavity point per concavity region is detected, which is not always correct, because there can be clumps that have more than one concavity point in a particular concavity region. The clump in Figure 1 is a simple case in which there are two concavity points in the concavity region R_1, shown as small white squares. Obviously, in order to find the accurate split lines using a concavity point-based method, identification of all the valid concavity points in the initial phase is necessary.

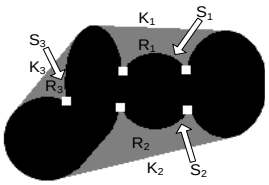


Figure 1. A clump of four convex objects (black) containing more than one concavity point (white) in two of its concavity regions (gray).

The literature contains only few fully satisfactory methods for concavity point detection. One of them uses curvature analysis [10]. The method first finds the value of curvature at every boundary point, and selects only those points as concavity points which have a curvature value larger than a predefined threshold. Consequently, all the valid concavity points are detected irrespective of how many concavity points belong to the same concavity region. Hence, an improvement is achieved over the approach used by Kumar *et al*.

2.2. Listing candidate split lines

As the number of concavity points increases, the number of possible split lines also increases. For N concavity points the number of possible split lines is $\binom{N}{2}$. Typically most of these split lines are invalid or intersect other split lines, and it is necessary to exclude them from further analysis. We take the approach from [10] and use Delaunay triangulation (DT) for this purpose. The properties of DT that it disallows the lines to intersect each other and that it maximizes the minimum of the interior angles of the triangles formed between the split lines lead to its usefulness in finding a list of possible split lines. The split lines in this list are referred to as candidate split lines.

2.3. Finding the best split lines

The third and final step is finding the best split lines from the list of candidate split lines. In order to do that, a set of features is extracted from the image for all the concavity points in the list of candidate split lines. These features are then used to define the best split lines. These include saliency, concavity-concavity alignment (CC), and concavity-line alignment (CL) features taken from [4], but they are modified to remove the deficiencies present in them.

Intuitively, a valid split line should be as short as possible, and it should also have concave enough regions at both of its ends. Saliency is the feature that ensures that these requirements are met by taking into account the concavity depth of the concave regions at the concavity points as well as the distance between the two concavity points forming the split line. The concavity depth (CD) is the perpendicular distance between the concavity point and the convex hull chord of the concavity region to which the concavity point belongs. Kumar *et al*. came up with an expression for saliency as the ratio of the minimum of the two concavity depths ($\min(CD_i,CD_j)$) to the sum of this minimum and the Euclidean distance between the two concavity points.

(a) (b)

Figure 2. In the original method by Kumar *et al.*, selection of a large enough saliency threshold to make the valid split line in (a) causes the invalid split line in (b).

However, we observed a problem with this expression: as the value of $\min(CD_i,CD_j)$ increases, then for a particular saliency value, the allowed distance between two concavity points also increases. This makes it very difficult to find a global threshold value for saliency, with the requirement that the threshold should work well for the whole image set. For example, if $\min(CD_i,CD_j)$ is small even though the respective split line is valid, even a small value of the distance between the two concavity points makes the saliency small (see Figure 2(a) where the split line is valid). But due to the nonlinear increase in distance w.r.t the decreasing saliency values, putting a very small global threshold for saliency would lead to the acceptance of long split lines, in the cases where the $\min(CD_i,CD_j)$ values are large (see Figure 2(b) where the split line is invalid).

We modified the definition for saliency such that the variation in distance for a particular saliency value is kept small when $\min(CD_i,CD_j)$ increases. Our tests on clumps present in the images used for this study indicate that this can be achieved by using the squared values of both of the parameters in the denominator. The modified expression for saliency is

$$SA_{i,j} = \frac{\min(CD_i, CD_j)}{0.1 \times \min(CD_i, CD_j)^2 + D_e(C_i, C_j)^2}, \qquad (1)$$

where CD refers to the concavity depth and $D_e(C_i,C_j)$ is the Euclidean distance between the two concavity points. A large value for $SA_{i,j}>0$ (below unity for clumps observed in this study) is expected for valid split lines.

A valid split line is typically characterized by having good alignment between the concavity regions to which the two concavity points belong. CC and CL alignment features take this into consideration. Both of these alignment features are based on the orientation of the two concavity regions, which is characterized by a unit vector. Therefore, a slight error in the definition of the unit vector causes inaccurate values for CC and CL.

The definition for the unit vector that is given by Kumar *et al*. is that it is a unit length vector that originates from the midpoint of the corresponding convex hull chord towards the concavity point. This definition of unit vector may be reasonable for the case in which a concavity region has only one concavity point. However, as we already mentioned above, many complex clumps have concavity regions that have more than one concavity point in them. Therefore the Kumar *et al*. definition of

unit vector is not valid. Furthermore, sometimes the shape of a concavity region is such that even if it has only one concavity point, the unit vector obtained by the Kumar *et al*. definition does not accurately describe the orientation of the concavity region.

We therefore require that the unit vector bisect the region in the proximity of the concavity point rather than the whole concavity region from the convex hull chord. This is illustrated in Figure 3.

In order to get the correct unit vectors, we developed a new approach in which we first trace the contour of the cell clumps in the clockwise direction using the concept of chain codes. As we move along the contour of the cell clump, we make a linked list of spatial coordinates of the contour with their index values. Therefore, for a particular coordinate value its index can be obtained, and hence the neighboring contour points can be accessed as well, through incrementing or decrementing the index values.

In order to find the unit vector of a concavity point, first its index value is obtained. Then, using that index value, two points at a predefined distance on the contour are taken, one on either side of the concavity point, and the midpoint of the straight line connecting those two points is found. This point is then used as the tail-point of the unit vector. The head of the vector of course is put on the line that connects this point and the concavity point. This way the obtained unit vectors conform to the natural condition mentioned in the previous paragraph.

Once we have found the unit vectors, we can determine the CC and CL alignment features. CC alignment is the angle which describes the degree of opposite alignment of the two concavity regions, and is given by

$$CC_{i,j} = \pi - \cos^{-1}(v_i \cdot v_j), \qquad (2)$$

where v_i and v_j are the unit vectors of the two concavity points. CL alignment describes how well the two regions are aligned with the split line and is given by

$$CL_{i,j} = \max(\cos^{-1}(v_i \cdot u_{ij}), \cos^{-1}(v_j \cdot u_{ji})), \qquad (3)$$

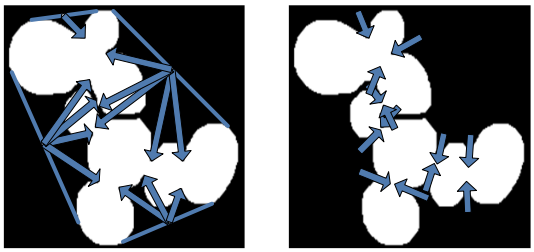
where u_{ij} is the unit vector along the line from point i to point j. For a valid split line, both $CC_{i,j} \in [0,\pi]$ and $CL_{i,j} \in [0,\pi/2]$ alignment features are desired to be minimal.

The next step is to evaluate a cost function for every point-pair using these extracted features. The cost function that we use is defined by

$$CF_{i,j} = SA_{i,j} + CC_{i,j} + 2 \times CL_{i,j}. \qquad (4)$$

The logic behind multiplying $CL_{i,j}$ by two is that the range of $CL_{i,j}$ is half that of $CC_{i,j}$.

Finally, each concavity point is paired with the point that gives the smallest value of the cost function. This defines the best split line for that concavity point. If this results in a point-pair where both points have already been used in another split line, the pair is not used to define a new split line. Finally, the obtained split lines are applied to the binary image to split the cell clumps.



(a) (b)

Figure 3. (a) Directions of the unit vectors obtained by using the original method by Kumar *et al.*[4]. (b) Valid directions of the unit vectors found by the modified method.

3. RESULTS

3.1. Test images and initial segmentation

We tested the clump splitting methods on images of yeast cells triply stained with FITC-ConA, taken from the Saccharomyces Cerevisiae Morphological Database (SCMD) [7]. We took a random sample of 35 images containing 1080 yeast cells that belonged to cell clumps. The images were segmented by a local thresholding method that is based on the classic threshold selection method by Otsu [6]. Details on the segmentation method can be found on the supplementary website at http://www.cs.tut.fi/sgn/csb/imclump/.

3.2. Comparison of methods

We evaluated the performance of the improved clump splitting method (IM), method by Kumar *et al.* (KM), and the classic watershed-based method (WS) on the segmented images containing yeast cells of approximately circular and elliptical shapes. The watershed-based method operates on the distance transformed binary image, see e.g. [8] or the supplementary website. We obtained true positives (TP), false positives (FP), and false negatives (FN) by manually going through the results of these methods and obtained precision (PR) and recall (RC) measures, see Table 1, by

$$PR = \frac{TP}{TP + FP}; \quad RC = \frac{TP}{TP + FN}. \qquad (5)$$

A compact representation of the segmentation accuracy is obtained by using the F-measure (FM) from [2], see Table 1. The F-measure is the harmonic mean of the precision and recall measures and is given by

$$FM = \frac{2}{1/PR + 1/RC}. \qquad (6)$$

It is clear from Table 1 that IM is superior to KM as well as WS on the basis of the F-measure. The precision for IM is slightly smaller than precision for KM and WS. However, IM has a considerable gain over KM and WS in the recall value. Figure 4 illustrates the superiority of IM over KM and WS using an image taken from the set of images used in [5]. Note that this image is not one of the 35 images used for Table 1.

Table 1. Performance parameters of the methods for 35 images containing 1080 yeast cells. (See text for the abbreviations.)

	TP	FP	FN	PR	RC	FM
KM	807	11	273	0.987	0.747	0.850
WS	859	1	221	0.999	0.795	0.886
IM	986	21	94	0.979	0.913	0.945

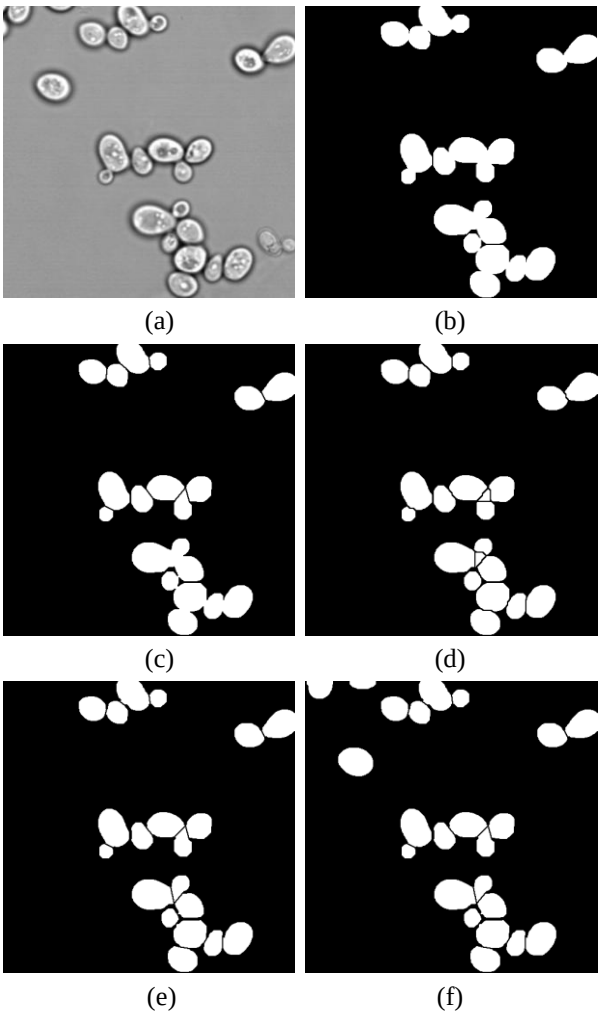


Figure 4. (a) A bright field image of yeast cells. (b) Segmented image showing only clumps. Resulting image after application of (c) Kumar *et al.* method, (d) watershed-based method, and (e) improved method on the image in (b). (f) Result of improved method with non-clumped cells included.

It is clear from Figure 4 that KM under-splits some clumps because of taking only one concavity point in one concavity region and also due to the wrong definition of unit vectors. On the other hand, WS over-splits some clumps. In contrast, IM splits all the clumps correctly. We observe similar behavior with the images obtained from the SCMD database as well. The images taken from the database as well as the respective clump splitting results can be seen at the supplementary website at http://www.cs.tut.fi/sgn/csb/imclump/.

4. CONCLUSION

In this paper, we presented an improved concavity point-based clump splitting method. Qualitative and quantitative comparisons show that it performs better than the compared clump splitting methods. The method was tested with clumps of yeast cells, but it can be applied to clumps of other convex objects as well. In future work we will consider post-processing methods that can improve clump splitting results obtained by all of the three methods considered in this paper. Completely new methods for clump splitting will be considered as well.

5. ACKNOWLEDGMENTS

This work was supported by the Academy of Finland (application number 213462, Finnish Programme for Centres of Excellence in Research 2006-2011; application number 121830, Post-Doctoral Researcher's Project 2008-2010). The SCMD database has been provided freely by the University of Tokyo for use in this publication only.

6. REFERENCES

[1] D. Balthasar *et al.*, “Real-time detection of arbitrary objects in alternating industrial environments,” in *Proc. 12th Scandinavian Conf. Image Analysis*, Bergen, Norway, June 11-14, 2001, pp. 321–328.

[2] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recogn. Letters*, vol. 27, no. 8, pp. 861–874, June 2006.

[3] G. Fernandez *et al.*, “A new plant image segmentation algorithm,” in *Proc. 8th Int’l Conf. Image Analysis and Processing*, Italy, Sept. 13-15, 1995, pp. 229-234.

[4] S. Kumar *et al.*, “A rule-based approach for robust clump splitting,” *Pattern Recogn.*, vol. 39, no. 6, pp. 1088-1098, 2006.

[5] A. Niemistö *et al.*, "A K-means segmentation method for finding 2-D object areas based on 3-D image stacks obtained by confocal microscopy," in *Proc. 29th Annual Int'l Conf. IEEE Engineering in Medicine and Biology Society*, Lyon, France, August 23-26, 2007, pp. 5559-5562.

[6] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62-66, 1979.

[7] T. L. Saito *et al.*, "SCMD: Saccharomyces cerevisiae morphological database," *Nucl. Acids. Res.*, vol. 32, pp. D319-D322, 2004.

[8] P. Soille, *Morphological Image Analysis: Principles and Applications*, 2nd ed. Heidelberg, Germany: Springer Verlag, 2003.

[9] R. J. Taylor *et al.*, “Dynamic analysis of MAPK signaling using a high-throughput microfluidic single-cell imaging platform,” *Proc. Nat’l Acad. Sci. USA*, vol. 106, no. 10, pp. 3758-3763, 2009.

[10] Q. Wen *et al.*, “A Delaunay triangulation approach for segmenting clumps of nuclei,” in *Proc. 6th IEEE Int’l Symp. Biomedical Imaging*, Boston, Mass., USA, June 28 - July 1, 2009, pp. 9-12.

[11] W. Wang *et al.*, “Cell cluster image segmentation on form analysis,” in *Proc. 3rd Int’l Conf. Natural Computation*, Haikou, China, Aug., 2007, pp. 833-836.

[12] C. Wählby *et al.*, “Algorithms for cytoplasm segmentation of fluorescence labelled cells,” *Anal. Cell. Pathol.*, vol. 24, no. 2-3, pp. 101-111, 2002.

Publication IV

M. Farhan, O. Yli-Harja, and A. Niemistö, "A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search," *Pattern Recognition*, vol. 46, no. 3, pp. 741-751, March 2013.

A Novel Method for Splitting Clumps of Convex Objects Incorporating Image Intensity and Using Rectangular Window-Based Concavity Point-Pair Search

Muhammad Farhan, Olli Yli-Harja, Antti Niemistö

Department of Signal Processing, Tampere University of Technology,
P.O. Box 553, FI-33101 Tampere, Finland,
muhammad.farhan@tut.fi, olli.yli-harja@tut.fi, antti.niemisto@tut.fi

Corresponding author: Muhammad Farhan

Email Address: muhammad.farhan@tut.fi

Phone: +358 40 4652111

Fax: +358 3 3115 4989

Postal address: Department of Signal Processing, Tampere University of Technology, P.O. Box 553, FI-33101 Tampere Finland.

Abstract

A novel nonparametric concavity point analysis-based method for splitting clumps of convex objects in binary images is presented. The method is based on finding concavity point-pairs by using a variable-size rectangular window. The concavity point-pairs can be either connected with a straight split line or with a line that follows a path of minimum or maximum intensity on an accompanying grayscale image. Using straight lines can result in non-smooth contours. Therefore, post-processing steps that remove acute angles between split lines are proposed. Results obtained with images that have clumps of biological cells show that the method gives accurate results.

Keywords: *Image segmentation, clump splitting, intensity-based splitting, concavity point, split line*.

1. Introduction

In digital imaging domain, it is often observed that the objects in an image clump together. This might occur due to high density of objects in an image area or objects being extremely close to each other that due to optical projections the objects in the image appear to be touching each other and forming clumps [1]. In some application domains, the objects in a scene being imaged might actually overlap and form clumps, for example, the image of objects moving on a conveyor belt [2]. Our target application area is microscopic imaging of biological cell cultures, where clumping of cells occurs naturally. This is because some cell types, such as yeast cells, and many different bacteria have the tendency to grow in clumps.

In these imaging applications and also in many applications in the field of computer vision, see for example [3] and [4], accurate automated image analysis requires that the clumps are split into their constituent objects. For example, it is necessary to extract single cells from an image in order to study the dynamics of single-cell gene expression [5]. As the objects forming clumps usually have similar intensity values and often inconspicuous edges, the general image segmentation methods fail to separate the individual objects from the clump. Therefore, in high-throughput automated image analysis involving such images, a post-processing step of clump splitting is typically performed after the initial segmentation.

Many of the clump splitting methods found in the literature assume the objects in the image to be convex, see for example [3, 6-11]. Using this assumption they try to find specific points, called concavity points, on contour segments where the object ceases to be convex. Clump splitting is then achieved by joining pairs of such points. When the clumps are complex, it is common for these methods to suffer from under-splitting. Another problem with these methods is that they depend on several user-defined parameters to get the pair for a concavity point. This causes their performance to degrade since it is difficult or even impossible to optimize the parameters to get high overall split accuracy when the image set is large with varying object sizes, shapes and the extent of their overlap. Another important issue that is not addressed well in earlier methods is the fact that there tends to be some holes within the clump when the number of objects in the clump increases. Under-splitting occurs if they are not taken into account while finding the split lines. Moreover, none of the earlier methods take due advantage of the intensity values of the image in order to split the clumps more accurately.

Here in this paper, we present a comprehensive method for splitting clumps based on the concavity point analysis, taking the aforementioned issues into account. We propose a new method for the detection of concavity points. The method uses the definition of convexity to find the maximum curvature points from concave region of the contour. We also propose a novel method which uses variable-size rectangular window to search for the best concavity point-pair. With this technique the dependency on the user-defined parameters is reduced along with an increase in the segmentation accuracy. Also, we have incorporated the prominent corner points on the contour of holes inside clumps to get the complete set of split lines.

Moreover, we have developed an algorithm to follow the minimum/maximum intensity path between two points, to be used for the images where the intensity values can be used as a clue to split the clump. In addition, we present a post-processing technique to be employed in the case when the intensity values cannot be used for finding the split line. It removes the residual objects or the ones that do not conform to the objects presumed in the image based on *a priori* knowledge about the object shapes.

The rest of the paper is organized as follows: In Section 2, we present a review of state-of-the-art clump splitting methods. Section 3 presents the proposed method whereas Section 4 describes the proposed post-processing technique. In Section 5, we present and discuss the quantitative and qualitative results obtained by

applying the proposed method and two other methods reviewed in Section 2 on the synthetic images of cell populations as well as on cell microscopy images. Section 6 concludes the paper.

2. Review of concavity point analysis-based methods

The concavity point analysis-based methods mostly utilize a general step-wise procedure, such as, detecting concavity points, finding candidate split lines and choosing the best split lines [3, 6-13]. An alternative approach is to use concavity points to segment the object contour and to fit ellipses to the contour segments to split the clumps [14-16]. Here, we highlight the deficiencies of those concavity point analysis-based methods which by far produce the best results to the best of our knowledge. Moreover, most of these deficiencies have been rectified in the method proposed in this paper.

The method by Kumar et al. [8] fails to find all concavity points when there are multiple concavity points in a concavity region. Moreover, its expression for saliency, a feature used to shortlist candidate split lines, gives a highly nonlinear relationship between the depth of concavity and the length of split lines. This results in long invalid split lines when allowing a reasonable length split line for a concavity point with less concavity depth. Also, the directional vector used to give the orientation of a concavity region is defined in such a way that in many cases it does not match the perceived orientation of the concavity. This method is modified by us in [6] to achieve improvement in the problematic areas. However, it suffers from parameter dependency and also results in over-splitting along with producing objects irrelevant to the actual objects present in the image.

The method by Wang in [3] uses polygonal approximation to smooth the object contour. This may deform the shape of the objects, and can even result in a loss of concavity points which have small concavity depths. The method picks some significant concavity points as candidate points and splits the clump from them. However, as the objects are split, some of those significant concavity points disappear, causing loss of potential split lines between them and some non-significant points. To make a split line the method poses the requirement that the concavity region of the second concavity point should lie within the cone formed by the extension of the vertices of the first concavity point towards it. However, using the cone can be misleading in cases where the angle between the vertices is small or there are two concavity points in the cone. In the latter case, the method prefers the concavity point with higher degree. However, the length of the split line is a

much more significant parameter to decide between the split lines. The method performs morphological opening of holes through a minimum distance path found between the corner of the hole and a point on the object contour. However, not all the corner points of a hole should have a split line through them, also the minimum distance path may not yield the optimal split line.

The approach used by Liang in [9] for detection of concavity points causes invalid concavity points since thresholding the angle near concavity alone is not a good criterion without considering the depth of the concavity. It accepts the shortest possible path of lowest possible gray values provided that the ratio of the length of the large and the small object contours is less than a predefined threshold. However, this may lead to false split lines because a path of lowest possible gray values between a concavity point and a contour point is not optimal unless it is found using some directional search.

The method in [13] finds concavity points on the basis of distance between potential concavity point-pairs, from inside and along the contour, and not on the basis of concaveness. However, sometimes a point that is somewhat further is the best pair for a concavity point rather than the nearest point. The method then finds a split path in the intensity patch formed by a rectangular window between concavity points. However, such a split path based on image intensity is usually a curve which tends to go outside that window, and therefore a directional search is needed to find the path. The method in [12] uses watershed segmentation to get initial clump splitting and then eliminates false split lines resulting due to over-splitting. Then it applies concavity point analysis-based clump splitting which is similar to the method in [8] and has many of the same issues.

The methods based on concavity points and ellipse fitting [14-16] start with performing polygonal approximation of the contour and then detect concavity points by using the angle between the vertices or the changing angle of tangents to the contour. The concavity points are used to segment the contour and ellipse fitting is performed on those contour segments. However, these methods are typically computationally complex and parameter-dependent [8, 17]. Moreover, the ellipse fitting is not able to split complex clumps into individual objects because of the absence of contour segments inside the clumps and due to unknown number of objects in the clump. Moreover, when the image set contains objects of varying shapes and sizes, these methods may not be able to perform accurately.

3. Method

This section describes the proposed novel clump splitting method, the steps of which are delineated by the flowchart in Figure 1. The method operates on binary images obtained after initial segmentation; however, intensity values of the image can also be used as additional information for finding the split lines. The method attempts to separate all the individual objects in a clump at once. However, it is iterative in the sense that it repeats clump splitting on the objects that could not get split in the initial phase. The advantage of performing splitting at once, besides being faster, is that a concavity point can have more than one split line through it, whereas when each split line is considered one by one, once a split line is drawn then another potential split line may be lost. This happens because the objects get separated and the two points involved in that split line do not exist anymore so as to be considered as pairs for some other concavity points.

FIGURE 1

3.1. Image Pre-processing

Sometimes there exist holes within the clumps of objects which are formed due to the clustering of several objects together, Figure 2(a) shows an example of such a case. In order to accurately split such clumps, prominent points on their contour (for example, blue squares in Figure 2(b)) should be paired with other such points or the concavity points (for example, red squares in Figure 2(a)) to form the split lines. All such holes and their corresponding prominent points should be found in the very beginning so that during the pair-search for a particular concavity point those points are also considered. The prominent points are found by analyzing the points on the contour segments of holes. Within a particular contour segment, the contour point having the largest distance from its corresponding imaginary local chord is the desired point provided that the midpoint of that chord lies on the background.

FIGURE 2

3.2. Concavity point detection

The next step is to detect all those points on the contour of the clumped object which are the points of intersection of two touching objects. Since it is assumed in the later step that every obtained concavity point is valid, concavity point detection needs to be performed carefully such that no single point that is taken was a

result of boundary irregularities. A predefined minimum concavity depth value is employed to serve this purpose. There are several methods [7, 9-11] in the literature that are used to detect the concavity points in clumped objects. However, they often fail to determine all the concavity points present in a clump. Here we develop a new technique which is very simple and effective, and is based on the definition of convexity.

The idea is to take two contour points and imagine a line between them, see for example, blue or yellow lines in Figure 2(a) where green circle indicates the initial point of the line. Taking too distant or close points may cause failure in detection of valid concavity points. However, it is observed through experiments with different images that a line between the end points of a 20 pixels long contour segment is applicable to clumps of objects of any shape with the contour length greater than 20 pixels. If that entire line resides inside the object (such as blue lines in Figure 2(a)), the convexity of the object is assured along that segment of the contour. In contrast, if the line passes through the background (such as yellow lines in Figure 2(a)) then there lies a concavity point along that segment of the contour. In this latter case, the next step is to find the distance of each of the points on that contour segment with their respective imaginary local chords(such as red line in Figure 2(a)) such that the midpoint of the respective chord lies on the background, otherwise that particular point is ignored. The imaginary local chord is obtained by joining sixth adjacent contour point on either side of the current point. Finally the point which gives the maximum of the distance is selected as the concavity point provided the distance value is larger than or equal to 2 pixels ($\sqrt{2}$ being the distance between diagonal pixels).

In the case of satisfying the convexity criterion, the third adjacent point to the previous first point is taken as the initial point, whereas if the convexity criterion is not satisfied, the third adjacent point to the previous second point is taken as the initial point of the next contour segment. Again a 20 pixel long contour segment is taken and the process is repeated until the starting point is reached. The concept is illustrated in Figure 2(a).

3.3. Concavity point-pair search

There are $\binom{N}{2}$ possible split lines for a clump with *N* concavity points. Even if the intersecting lines and the lines passing through the background are omitted, the remaining split lines may not necessarily be all valid. Therefore, the preferred approach is to find the best split line or the concavity point-pair for each of the

concavity point in the clump, rather than analyzing every possible split line. In a previous method in [6], we used different features to choose the best split line for every concavity point from the list of possible split lines. However, we observed that there are cases in which it is very difficult to decide between two possibilities of the best split lines because of using the same parameters for the whole image set.

Thus in order to focus on finding only the best split lines individually for each concavity point and also to eliminate user-defined threshold values, we take into consideration the fact that the split lines should be found within a specific region along the directional vectors (purple arrows in Figure 3) associated with the concavity point. The directional vector should ideally bisect the region in the vicinity of the concavity point. This is realized by defining it as a vector with its tail on the midpoint of the imaginary local chord (red line in Figure 2(a)), corresponding to that concavity point, and originating towards the concavity point. This concept leads us to start searching for the pair of a concavity point in the area along and on either side of its directional vector giving rise to a variable-size rectangular window as illustrated in Figure 3.

FIGURE 3

The idea is that first the directional vector associated with a concavity point is found. Then, two points are picked on the object contour, one on either side of the concavity point and both equally distant to it, and a line is formed between them. This line is then extended on either side using those two points along with the directional vector. The length of this line is referred to as width of the window *w*. The reason behind extending the line is that when the region around a concavity point is narrow and deep then the window formed by using the contour points would always be thin and at an undesired location, see for example, Figure 3(a). Thus formation of the window should be independent of how the contour progresses beyond the concavity point. Figure 3(b) illustrates the point where gradually increasing window width leads to the successful detection of the concavity point-pair (red window).

The other two corner points of the rectangular window are found at a distance *h* on the other side of the contour and in the direction of the directional vector using basic trigonometric relations. That distance *h* is a parameter which defines the length of the window and depends on the maximum length of the split line that is allowed in a certain image set. Once the coordinates of the four points are obtained, all the pixels within the rectangular area bounded by them constitute a rectangular window. This window is then used as a mask to

search for the pair of the concavity point under consideration. The idea here is to use variable size window. Therefore, initially a small window width and a comparatively large window length are chosen based on the prior knowledge about the object size in the image set. The window width is gradually increased until a concavity point is found inside the window or the window width approaches its maximum value.

The purpose of using small window width at the beginning is to avoid the case of having two concavity points inside the search window. Intuitively, split lines must be as short as possible; therefore, even if there exist two concavity points in the window, the concavity point having the smallest distance from the subject concavity point is accepted as its pair. If the window width reaches its maximum value and no concavity point is found inside the window, then the window length is iteratively increased until a concavity point is found inside the window or the maximum window length is reached.

This whole process is repeated for every concavity point present in the clumped object and a list of the concavity point-pairs is formed. Next, any such concavity point is found that was left without being assigned a pair. Often the pairing concavity point for such a concavity point was discarded in the initial concavity point detection phase due to lack in concaveness or due to boundary irregularities. In such cases and in the case when there is only one concavity point in a clump, a line is drawn from that concavity point to a point on a segment of object contour in the direction of the directional vector associated with the concavity point. The point is chosen from a contour segment such that, among a certain number of points in that segment, it has maximum distance from its corresponding local chord provided the midpoint of the chord lies on the background. This point-pair is also added to the list of concavity point-pairs.

During this whole process, the prominent points on the contour of the holes are also considered while searching the pairing point for a particular concavity point. However, there are cases in which the clumps are so complex that there are many such holes lying inside them. In such cases, often a split line is also realized by joining a pair of prominent points belonging to two different holes lying nearby. Therefore, like individual concavity points, the pairing points for all the prominent points of all the holes of an image are also found so that all valid split lines are identified.

3.4. Split line formation

Once the concavity points are obtained, there can be two different approaches for obtaining the split lines: making a list of concavity point-pairs, as described in the previous section, and joining them through straight lines, or finding a path of minimum/maximum intensity from a concavity point to a point in another concavity region or to an already drawn split line. The former approach is appropriate when the image intensity values cannot be used as conclusive evidence for determining the split path. Even though this approach may separate the clumps into their correct number of constituent objects, it may not give the correct individual object areas. Moreover, for some complex clumps straight lines can produce erroneous results, since for a given set of images long split lines may be allowed, but making such long straight lines may not match the underlying objects despite being algorithmically correct, see for example Figure 4(c).

FIGURE 4

Furthermore, this approach may lead to under-segmentation. This can be avoided by using an iterative procedure in which it is checked if the result of the first round of clump splitting yielded larger objects. A scaled value of the constraint for the smallest allowed object is used to decide if an object requires further processing. Clump splitting is then performed iteratively on such objects while maintaining that the over-splitting is not achieved.

When the original image has discernible intensity variations along the region where the objects seem to merge together, see for example Figure 4(a), then we should rather find the minimum/maximum intensity path to effectively split the clumps than opting for a straight line between concavity point-pairs. Here, we deduced an algorithm which finds the minimum/maximum intensity splitting path using the directional vector associated with the concavity point which guides the algorithm in the right direction and prohibit it from straying.

FIGURE 5

Here, we use a 3x3 mask centered at the current point, starting from the concavity point, to locate the next low/high intensity valued pixel in the intensity image. Depending on the direction of the directional vector associated with a concavity point, one of the four 3x3 search masks, illustrated in Figure 5(a), is used. For example, if the angle of the directional vector with respect to the horizontal line is in the range $0 < \theta \leq \pi/2$

then the top-right mask is used. Similarly, for the case, $\pi/2 < \theta \leq \pi$, the top-left mask is used and so on. Notice the equality and inequality condition while choosing the mask as otherwise the search might go in the wrong direction. Since the directional vector remains fixed, once a search mask has been chosen for a concavity point, it is used unchanged. Now, if this new point with lowest/highest intensity in the 3x3 neighborhood does not correspond to a background pixel in the binary image, then it is assigned the background pixel value and made the current point. The center of the mask is put on it and the procedure is repeated until a point is reached which corresponds to a background pixel in the binary image.

The end point of the split line found in the procedure is compared with the points of all the concavity regions for the clumped object, and only if it is part of one of them or part of the image border then the line can be retained, otherwise it is discarded. This ensures that the line is made between the subject concavity point and a point on the concavity region in the direction of the directional vector or with a point at image border. In any case, the size of each objects resulting from this new line must be larger than the value for the smallest allowed object in the image, otherwise the line is discarded. In this way, the obtained split lines resemble a lot to the lines that an expert would draw. Consequently, it helps in splitting complex clumps more accurately with better realization of actual individual object areas. Figure 4(d) illustrates the case where the usage of intensity information results in accurate clump splitting, thus giving the actual object areas as compared to the ones obtained by splitting using straight split lines in Figure 4(c). Figure 5(b) shows an example of using the mask (top right mask in Figure 5(a)) to find the path of minimum intensity (black curve in Figure 5(b)) from a concavity point to another one, which is a curve rather than the straight line (red line in Figure 5(b)). Colors along the minimum intensity path refer to steps of the path finding procedure.

4. Post-processing technique

When the intensity information is not used, the clump splitting method described in the previous section defines mere straight lines between the concavity point-pairs without considering the relationship that can exist between the split lines. For example, sometimes there are two split lines through a particular concavity point making an acute angle between them, as shown in Figure 6(a). Moreover, sometimes the other two concavity points involved in those two split lines also share a split line between them which results in a

triangular object in an image, as illustrated in Figure 6(b). If there is prior information about the object shapes and also the objects are known to have smooth boundaries, then both those cases lead to an output image with objects not corresponding to the topology of the underlying image objects. Moreover, in the latter case there is an extra object which is not in accordance with the shape of the objects actually present in the image. Therefore, we need to post-process the resulting image from the initial clump splitting to make the final split lines mimic the manually obtained split lines.

FIGURE 6

Here, we propose a post-processing technique to solve these cases. The process begins with finding the two cases by checking the degree of all the concavity points present in the object. The term degree is used here to specify the number of split lines passing through a concavity point. By going through the list of concavity point-pairs, such concavity points are found whose degree is two. Then the other two concavity points are taken which share the split lines with the first concavity point and it is checked if there exist a split line between them or they do not share any split line with any other concavity point. Once either of these conditions get fulfilled then a triangle is formed between the three points, if it was not already there, and the centroid of that triangle is found. After finding the centroid, the concavity point-pairs formed by those three concavity points are removed from the initial list and are replaced with three point-pairs each involving the centroid and one of the three concavity points. Figure 7 shows the output of the post-processing step for our example cases of Figure 6.

FIGURE 7

Sometimes it might also happen that a concavity point has three split lines passing through it, see for example Figure 8(a). In such a case, those three lines can be perceived as two pair of lines emerging from that concavity point. Then the pair of lines which give smaller of the two angles are analyzed. Associated with those two split lines are the two other concavity points. If the degree of only one of those two concavity points is two then the split line involving that concavity point is discarded from the list. If the degree of both the concavity points is two then the line involved in the wider of the two angles is discarded. If neither of the two concavity points have degree two then the normal post-processing is performed one after the other for the two pair of lines. Figure 8(b) shows the result of the post-processing applied on the image of Figure 8(a).

FIGURE 8

5. Results and Discussion

5.1. Image acquisition and Benchmark image set generation

Validation of image analysis methods is traditionally performed by comparison of results obtained by them with the ground truth created by manual analysis. Manual creation of ground truth, however, is time consuming, laborious and observer-dependent, especially in the case of high-throughput microscopic image analysis where we have very large sets of images. Therefore, the manual validation becomes impractical. Instead a benchmark set of synthetic images having varying properties mimicking the microscopic images, like the ones generated by SIMCEP tool [18] can be used.

Here we use both real microscopy images as well as synthetic images to evaluate our method. We have two different test cases based on whether the intensity information is usable for splitting or not. For the case of utilizing image intensity for splitting, we use the microscopic images and the corresponding manually obtained ground truth results. For the other case we use benchmark synthetic image set for which the ground truth information is available. This can be considered in a way that the first set performs both qualitative and quantitative evaluation whereas the second set gives the quantitative measures of our method.

For the first test case, Case I, we first acquired two sets of bright field images of the budding yeast *Saccharomyces cerevisiae* cells of varying sizes and shapes with a Leica TCS SP2 microscope. For each image, z-stacks comprising 20 images were captured using a 100X oil immersion objective (NA 1.40) but only one of them is included in the test image set. This image was selected by first finding the best in-focus image from these z-slices by using the Tenengrad method [19]. Generally, a slightly out-of-focus image is chosen because of its assistance in giving better segmentation accuracy. Therefore, the image just below (about 300 nm) the best in-focus slice was selected into the test image set. Segmentation of the images was carried out using the method from [20].

Another set for Case I is obtained from Saccharomyces Cerevisiae Morphological Database (SCMD) [21]. The set contains fluorescent images of budding yeast *S. cerevisiae* which provides an ideal scenario for testing the method against varying object sizes. The image set contains more than 300 images but since the analysis is to be performed manually, due to unavailability of ground truth, we chose to use just the first 40

images for obtaining quantification measures. Segmentation of the images is carried out by a local thresholding method that is based on the classic threshold selection method by Otsu [22].

TABLE 1

For the second test case, Case II, that is to test and validate the proposed method, we used the benchmark set of synthetic images of cell populations with realistic properties generated with the SIMCEP simulation tool [18]. They were generated using the package of files, downloadable from: http://www.cs.tut.fi/sgn/csb/simcep/, and by varying the parameters. The entire set contains simulated images of cell populations with the corresponding ground truth images in which the cells are represented as binary markers which are used for validation. The idea here is to create such image set which contains images with cell clumps of varying sizes and complexity. Therefore, the generated images consist of overlapping nuclei with three different values of clustering probabilities. For each clustering probability we used eight different values for the amount of overlap and simulated 50 images (altogether 1200 images constituting 24 image sets), each of which contains 200 cells with approximately 10 cell clusters per image. Table 1 shows the necessary parameters and their values to be used for generating the image set.

5.2. Performance evaluation parameters

The quantitative performance evaluation is performed using precision and recall analysis. We obtained true positives (TP), false positives (FP), and false negatives (FN) and the precision (PR) and recall (RC) are then obtained by

$$PR = TP/(TP + FP) \; ; \quad RC = TP/(TP + FN). \qquad (1)$$

A high value of PR implies that a high percentage of the objects detected by the method are actually the objects of the ground truth image. It decreases once the method detects objects not actually there in the ground truth. On the other hand, a high value of RC specifies that a high percentage of the objects of ground truth image are detected by the method. Furthermore, we use F-measure (FM) [23] which can be obtained by

$$FM = 2/(1/PR + 1/RC), \qquad (2)$$

and is considered to be a more robust measure of segmentation accuracy.

5.3. Results and discussion

The proposed method was applied on all the image sets of both the test cases. For Case I, that is, the case where image intensity is useful in splitting clumps, we performed manual analysis on all the image sets to obtain the TP, FP, and FN values and calculated the PR, RC and FM measures. The first two sets contained bright field microscopy images of yeast cells and the third set contained fluorescent microscopy images of yeast cells obtained from SCMD [21]. It is worth mentioning here that there is a difference between the images of the first set and the other two sets in that they provide the cases when the path of minimum and maximum intensity, respectively, is searched for between the two points to split the clumps. We additionally employed the non-intensity-based method (indicated by NI in Table 2) on these image sets to compare its performance with the one obtained by employing image intensity (indicated by I in Table 2). Table 2 lists the results for this test case.

TABLE 2

The results of applying the proposed method on a bright field microscopy image of yeast cells (an image from Set 1) are shown in Figure 9 whereas Figure 10 depicts the results of applying the proposed method on fluorescent microscopy image (an image from Set 3). The quantitative values of Table 2 manifest that the results obtained from the proposed method are accurate irrespective of whether or not the image intensity is used. Moreover, from Table 2, Figure 9 and Figure 10, it is clear that the proposed method gives better quantitative as well as qualitative results when the image intensity is used. However, it is also clear that the results from this approach depend on accuracy of the initial image segmentation. It is evident from both the Figures that the non-intensity-based method struggles in the situations when the clumps are either very complex or touch the image borders, causing some of the concavity points to be missed. However, its performance is still very promising in that it gives such quantitative measures which were not previously achievable.

For the second test case, we evaluated the proposed method using the benchmark image set for which we have the binary images containing masks representing the cells in the ground truth. In addition, we also applied the method from Kumar et al. [8] as well as our previous method [6] on these images to compare the proposed method against them. The analysis was performed by measuring the performance parameters after

applying these methods on the simulated images and comparing the results with the ground truth. The obtained performance parameters for 24 image sets are presented in detail in Table 3 where subscripts 1, 2, and 3 stand for proposed method, methods from Farhan et al. [6], and Kumar et al. [8] respectively. Each entry corresponds to the overall value for the set of 50 images (the number of cells is 10 000).

It is clear from Table 3 that our method outperforms the other methods proving its significance in resolving complex clumps. Table 3 shows that the proposed method performs accurately when the probability and the amount of overlap are small. In this case, the main deviation in the results is caused by the amount of overlap. When it is around 0.3 there is not much difference between results for different probabilities of overlap. However, when the amount of overlap increases from 0.35 the increase in probability of overlap causes degradation in the performance of the method. Nevertheless, the F-measure for the worst case (probability of overlap = 0.6, amount of overlap = 0.5) is 0.91, against 0.81 for our modified method and 0.58 for method by Kumar et al. The F-measure of 0.91can still be considered as high accuracy especially considering that accurate splitting of the clumps for such images with heavy overlapping is not always possible even for a human observer. Moreover, this performance may improve even further if image intensity can be used as the evidence of the split path as we have observed in the earlier case that the splitting of complex clumps is efficiently done when image intensity is used. Figure 11 shows a representative image from Case II and the result of applying the proposed method on it. It manifests the accuracy of our method in splitting complex clumps. However, it can also be seen that there is some artifact (top left quadrant of the Figure 11 (d)) which can be easily dealt with by using the image intensity information.

TABLE 3

It must be emphasized that, unlike the other methods, the proposed method does not need any user-defined parameters, nor does it depend on predefined threshold values of features. For example, the method in [8] requires threshold values for concavity depth, Saliency, CC and CL alignment etc. Similarly, the method in [3] requires threshold values for angles and lengths to find the degree of concavities for their classification. Moreover, the method in [9] uses threshold values for angle of concavity, length of split line, ratio of the longest to the shortest contour of objects resulting from splitting etc. However, in our method, since the window size is varied until a pairing point is obtained, or in the case of using image intensity, only the

directional vector of a concavity point is used to find its pair, so there is absolutely no need for any parameter values for point-pair selection. We only applied a minimum object size constraint to prevent the method from splitting smaller objects. Therefore, before applying the split lines to a clump it is ensured that they do not result in a smaller object otherwise the lines are discarded. The constraint also helps in deciding whether another phase of clump splitting is needed for larger non-convex objects. This constraint is obtained for every image set by looking at the size of the smallest allowed object in that set.

One of the problematic cases is also the clumps touching the image borders. In many image processing applications objects that touch the image borders are removed as a preprocessing step. However, we did not remove them, because, when images have large clumps (see for example Figure 11 mid- and bottom-left), that would result in many valid individual objects (cells) being removed unnecessarily. Another reason that we included the objects touching the image border is to show the promising results achieved by our intensity-based splitting. It is performing remarkably well in those areas where the concavity point is lost either due to complex clumps or due to being outside the image area because of having resulted from the objects touching the image border (see for example, in Figure 9 objects at bottom-right as well as in Figure 10 objects at top-right and bottom left). Therefore, our method processes such objects, and in most cases, even when the image intensity is not used, is able to resolve individual objects from such clumps that are away from the image borders. However, in the case of not using image intensity, due to the incomplete information near the borders of the image, object splitting is sometimes not accurate in those parts. In certain applications it might be reasonable to remove any objects that touch the image border after the clump splitting step. However, for the sake of showing the raw results from the clump splitting method we have not done so here.

FIGURE 9

FIGURE 10

FIGURE 11

6. Conclusion

We presented a novel non-parametric concavity point analysis-based clump splitting method which takes into account holes in the clumps and if possible the image intensity too to find the split lines. In the case of not utilizing image intensity, a rectangular window mask is used for finding the pairing points of a split line. This makes the method independent of user-defined parameters even if the image set is large and contains objects of variable sizes. A post-processing step using *a priori* knowledge about shape of the objects ensured that the final image contains objects conforming to the ones present in the actual image. Moreover, when the image intensity can be used as a clue for finding the split lines, a split line is obtained for every concavity point using its directional vector which guides the search for the splitting path of minimum/maximum intensity. The advantage of this approach is that it can accurately split complex clumps besides producing the output similar to the one obtained by a human observer.

Quantitative and qualitative measures illustrate the outstanding performance of our method for diverse sets of images having clumps of varying objects sizes and the probability and amount of overlap. Although the method is non-parametric in nature, a minimum object size constraint is used for a particular image set to restrain the method from splitting smaller objects. Even though the target application of the method was microscopy images containing clumps of cells of convex shape, it can be applied to a wide range of applications with images containing clumps of any convex objects.

7. Acknowledgements

This work was supported by the Academy of Finland (application number 213462, Finnish Programme for Centre of Excellence in Research 2006-2011; application number 121830, Post-Doctoral Researcher's Project 2008-2010). We are extremely thankful to anonymous reviewers for their critical evaluation and useful comments and suggestions which helped us improve the manuscript.

References

[1] O. Schmitt, M. Hasse, Radial symmetries based decomposition of cell clusters in binary and gray level images, Pattern Recognition 41 (6) (2008) 1905-1923.

[2] D. Balthasar, T. Erdmann, J. Pellenz, V. Rehrmann, J. Zeppen, L. Priese, Real-time detection of arbitrary objects in alternating industrial environments, in: Proceedings of the12th Scandinavian Conference on Image Analysis, 2001, pp. 321–328.

[3] W.X. Wang, Binary image segmentation of aggregates based on polygonal approximation and classification of concavities, Pattern Recognition 31 (10) (1998) 1503-1524.

[4] G. Zhang, D.S. Jayas, N.D.G. White, Separation of touching grain kernels in an image by ellipse fitting algorithm, Biosystems Engineering 92 (2) (2005) 135-142.

[5] R. J. Taylor, D. Falconnet, A. Niemistö, S.A. Ramsey, S. Prinz, I. Shmulevich, T. Galitski, C.L. Hansen, Dynamic analysis of MAPK signaling using a high-throughput microfluidic single-cell imaging platform, Proceedings of the National Academy of Sciences of USA 106 (10) (2009) 3758-3763.

[6] M. Farhan, O. Yli-Harja, A. Niemistö, An improved clump splitting method for convex objects, in: Proceedings of the 7th International Workshop on Computational Systems Biology, 2010, pp. 35-38.

[7] G. Fernandez, M. Kunt, J-P. Zryd, A new plant cell image segmentation algorithm, in: Proceedings of the 8th International 7Conference on Image Analysis and Processing, 1995, pp. 229-234.

[8] S. Kumar, S.H. Ong, S. Ranganath, T.C. Ong, F.T. Chew, A rule-based approach for robust clump splitting, Pattern Recognition 39 (6) (2006) 1088-1098.

[9] J. Liang, Intelligent splitting in the chromosome domain, Pattern Recognition 22 (5) (1989) 519-532.

[10] W. Wang, H. Song, Cell cluster image segmentation on form analysis, in: Proceedings of the 3rd International Conference on Natural Computation, 2007, pp. 833-836.

[11] Q. Wen, H. Chang, B. Parvin, A Delaunay triangulation approach for segmenting clumps of nuclei, in: Proceedings of the 6th IEEE International Symposium on Biomedical Imaging, 2009, pp. 9-12.

[12] Q. Zhong, P. Zhou, Q. Yao, K. Mao, A novel segmentation algorithm for clustered slender-particles, Computers and Electronics in Agriculture 69 (2) (2009) 118-127.

[13] H. Wang, H. Zhang, N. Ray, Clump splitting via bottleneck detection and shape classification, Pattern Recognition 45 (7) (2012) 2780–2787.

[14] X. Bai, C. Sun, F. Zhou, Splitting touching cells based on concave points and ellipse fitting, Pattern Recognition 42 (11) (2009) 2434–2446.

[15] G. Cong, B. Parvin, Model-based segmentation of nuclei, Pattern Recognition 33 (8) (2000) 1383-1393.

[16] S. Kothari, Q. Chaudry, M. D. Wang, Automated cell counting and cluster segmentation using concavity detection and ellipse fitting technique, in: Proceedings of the 6th IEEE International Symposium on Biomedical Imaging, 2009, pp. 795-798.

[17] M. Farhan, Automated Clump Splitting for Biological Cell Segmentation in Microscopy using Image Analysis, M.S. Thesis, Tampere University of Technology, Finland, November 2009.

[18] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, O. Yli-Harja, Computational framework for simulating fluorescence microscope images with cell populations, IEEE Transactions on Medical Imaging 26 (7) (2007) 1010-1016.

[19] Y. Sun, S. Duthaler, B.J. Nelson, Autofocusing in computer microscopy: selecting the optimal focus algorithm, Microscopy Research and Technology 65 (3) (2004) 139-149.

[20] A. Niemistö, T. Aho, H. Thesleff, M. Tiainen, K. Marjanen, M-L. Linne, O. Yli-Harja, Estimation of population effects in synchronized budding yeast experiments, in: Proceedings of the International Society for Optical Engineering, SPIE 2003. Image Processing: Algorithms and Systems II, 5014 (2003) 448-459.

[21] T. L. Saito, M. Ohtani, H. Sawai, F. Sano, A. Saka, D. Watanabe, M. Yukawa, Y. Ohya, S. Morishita, SCMD: Saccharomyces cerevisiae Morphological Database, Nucleic Acids Research 32 (2004) D319-D322.

[22] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man, and Cybernetics 9 (1) (1979) 62-66.

[23] T. Fawcett, “An introduction to ROC analysis,” Pattern Recognition Letters 27 (8) (2006) 861-874.

Figures

Figure 1 – Flowchart delineating the steps performed by our clump splitting method.

The different steps involved in getting the final clump split image. The input to the method consists of original intensity and its binarized image along with the binary images containing holes and labeled objects.

Figure 2 – Image depicting the scenario of clump with holes. Directional vectors and concavity points are also highlighted.

(a) A synthetic image with clump of objects having holes inside the clump. The process of finding concavity points on the left where all blue lines reside inside the object whereas yellow lines pass through the background indicating that the object ceases to be convex there. The points in such contour segments (for example, red squares) which give maximum distance from their local chords (red lines outside object contour in the top) are identified as concavity points. (b) Contour of the clumped object. Black and red arrows point towards the directional vectors associated with the object concavity point and prominent point of holes, respectively.

Figure 3 – Variable-size rectangular window-based concavity point-pair search.

Variable-size rectangular window-based concavity point-pair search. Orientation of directional vectors (purple arrows) associated with the top two concavity points. Rectangular window with varying width and length (shown with dashed blue, green and red lines) aligned in the same direction as the directional vector in order to search for the concavity point-pair. (a) Formation of window with points taken from object contour. (b) Formation of window irrespective of concavity region around the concavity point.

Figure 4 – Illustration of the usage of image intensity for finding split lines.

Illustration of the usage of image intensity for finding split lines. (a) Original bright field intensity image. (b) Segmented image. (c) Result of straight split lines. (d) Split lines obtained by employing image intensity.

Figure 5 – Masks and procedure to find the minimum/maximum intensity path for splitting.

Masks and procedure to find the min/max intensity path to get the split lines. (a) The angle formed by the directional vector associated with the particular concavity point is used to select the appropriate mask. Arrows inside indicate the range of angles corresponding to a particular mask. Black and red pixels are Don't care. (b) Procedure to find minimum intensity path for an example case where the directional vector is such that the top right mask was used to search for the path of minimum intensity (black curve) instead of taking straight line (red). Colors along the minimum intensity path refer to steps of the path finding procedure.

Figure 6 – Post-processing cases after initial clump splitting.

Post-processing cases. (a) Object with two split lines making acute angle between them. (b) A triangle is formed between the three concavity points.

Figure 7 – Result after post-processing for the case of two split lines through a concavity point.

Result after post-processing. The objects of Figure 6(a) and (b) after the application of image post-processing.

Figure 8 – Illustration of the case of three split lines through a concavity point and its post-processing.

(a) A clumped object with the case of three split lines through a concavity point. (b) Resulting image after image post-processing.

Figure 9 – Results of proposed clump splitting method for a bright field microscopy image containing clumps of yeast cells.

(a) A bright field image of yeast cells. (b) Segmented image. (c) and (d) Resulting image after application of the proposed method with and without using image intensity.

Figure 10 – Results of proposed clump splitting method for a fluorescence microscopy image containing clumps of yeast cells.

(a) A fluorescent image of yeast cells clumped together (obtained from SCMD [21]). (b) Segmented image. (c) and (d) Resulting images after application of the proposed method with and without using image intensity.

Figure 11 – Results of clump splitting method for a synthetic microscopy image containing clumps of cells. (a) A synthetic microscopy image generated from SIMCEP simulation tool with cell clustering probability = 0.5 and amount of overlap = 0.45. (b) Segmented image provided by the tool itself. (c) Binary image containing masks representing cells in the ground truth. (d) Resulting image after application of the proposed method on the image in (b).

Vitae:

Muhammad Farhan received the degree of Master of Science (MSC in IT) with distinction in 2010 from Tampere University of Technology (TUT), Tampere, Finland. He has worked as a Research Assistant at the Department of Signal Processing, TUT in 2009-2010. Currently he is working as a Researcher and a PhD student at the same department. His research interests include signal and image processing, biomedical image analysis, and pattern recognition.

Olli Yli-Harja received the degree of Doctor of Science (Technology) in computer science and applied mathematics in 1989 from Lappeenranta University of Technology, Finland. During 1988-1998 he was a research scientist at the Technical Research Centre of Finland, Helsinki University of Technology, and University of Helsinki. During 1998-2001 he was senior researcher at the Institute of Signal Processing in Tampere University of Technology and in 2005 a visiting scientist on University of Texas M.D. Anderson Cancer Center in Houston, Texas, USA. Currently he is a Professor in the Department of Signal Processing in TUT. His research interests include computational systems biology, image analysis, complexity and non-linear filters.

Antti Niemistö received the degree of Doctor of Science (Technology) in signal processing in 2006 from Tampere University of Technology, Tampere, Finland. He has been with the Department of Signal Processing at TUT since 1999. He visited The University of Texas M. D. Anderson Cancer Center in Houston, Texas, USA during 2003-2004. In 2007-2008 he was a Postdoctoral Fellow at the Institute for Systems Biology (ISB) in Seattle, Washington, USA. Currently he is a Research Fellow at the Department of Signal Processing in TUT. His research interests include biomedical image analysis and nonlinear signal and image processing. His current work focuses on developing image analysis methods for microscopy applications in cell and molecular biology, in particular for quantitative analysis of live cells in microfluidic devices.

Author Biography

Muhammad Farhan received the degree of Master of Science (MSC in IT) with distinction in 2010 from Tampere University of Technology (TUT), Tampere, Finland. He has worked as a Research Assistant at the Department of Signal Processing, TUT in 2009-2010. Currently he is working as a Researcher and a PhD student at the same department. His research interests include signal and image processing, biomedical image analysis, and pattern recognition.

Olli Yli-Harja received the degree of Doctor of Science (Technology) in computer science and applied mathematics in 1989 from Lappeenranta University of Technology, Finland. During 1988-1998 he was a research scientist at the Technical Research Centre of Finland, Helsinki University of Technology, and University of Helsinki. During 1998-2001 he was senior researcher at the Institute of Signal Processing in Tampere University of Technology and in 2005 a visiting scientist on University of Texas M.D. Anderson Cancer Center in Houston, Texas, USA. Currently he is a Professor in the Department of Signal Processing in TUT. His research interests include computational systems biology, image analysis, complexity and non-linear filters.

Antti Niemistö received the degree of Doctor of Science (Technology) in signal processing in 2006 from Tampere University of Technology, Tampere, Finland. He has been with the Department of Signal Processing at TUT since 1999. He visited The University of Texas M. D. Anderson Cancer Center in Houston, Texas, USA during 2003-2004. In 2007-2008 he was a Postdoctoral Fellow at the Institute for Systems Biology (ISB) in Seattle, Washington, USA. Currently he is a Research Fellow at the Department of Signal Processing in TUT. His research interests include biomedical image analysis and nonlinear signal and image processing. His current work focuses on developing image analysis methods for microscopy applications in cell and molecular biology, in particular for quantitative analysis of live cells in microfluidic devices.

Table 1 – Parameters to create benchmark synthetic image set from SIMCEP tool.

Parameter settings for creation of benchmark image set containing clustered nuclei with increasing clustering probability and amount of overlap.

Parameter	Value
Probability of clustering	0.4, 0.5, 0.6
Amount of overlap	0.15, 0.2,…, 0.5
Number of Image Sets	24
Images per set	50
Cells per image	200
Total number of cells for 1 set	10 000

Table 2 – Performance parameters obtained from the proposed clump splitting method for microscopy images.

Performance parameters for three image sets of Case I containing 740, 858 and 1242 total number of cells constituting clumps. (See text for abbreviations)

Set	TP	FP	FN	PR	RC	FM
Set 1_I	727	9	13	0.988	0.982	0.985
Set 1_NI	726	15	14	0.979	0.981	0.980
Set 2_I	841	6	17	0.993	0.980	0.987
Set 2_NI	826	8	32	0.990	0.963	0.976
SCMD_I	1219	18	23	0.985	0.981	0.983
SCMD_NI	1198	21	44	0.982	0.964	0.973

Table 3 – Performance parameters obtained from the clump splitting methods for synthetic images.

Performance parameters obtained after application of the proposed method (Subscript 1), Farhan et al. (Subscript 2) and Kumar et al. (Subscript 3) on 24 image sets generated from SIMCEP simulation tool each containing 10 000 total number of cells with or without clumps. Text in column "Set" is interpreted as PN_1ON_2 where P = probability, $0.N_1$ = probability of overlap, O = Overlap, $0.N_2$= Amount of overlap.

Set	PR_1	RC_1	FM_1	FM_2	FM_3	Set	PR_1	RC_1	FM_1	FM_2	FM_3
P4O15	0.999	0.995	0.997	0.985	0.858	P5O35	0.992	0.948	0.970	0.897	0.664
P4O20	0.998	0.992	0.995	0.972	0.802	P5O40	0.990	0.923	0.955	0.871	0.633
P4O25	0.996	0.986	0.991	0.951	0.761	P5O45	0.988	0.897	0.940	0.855	0.633
P4O30	0.995	0.974	0.984	0.935	0.726	P5O50	0.988	0.880	0.931	0.835	0.606
P4O35	0.994	0.960	0.977	0.909	0.698	P6O15	0.998	0.992	0.995	0.977	0.814
P4O40	0.992	0.944	0.968	0.890	0.673	P6O20	0.998	0.988	0.993	0.960	0.754
P4O45	0.990	0.922	0.954	0.870	0.664	P6O25	0.997	0.980	0.988	0.936	0.710
P4O50	0.987	0.896	0.939	0.851	0.652	P6O30	0.994	0.957	0.975	0.912	0.689
P5O15	0.998	0.993	0.995	0.982	0.842	P6O35	0.992	0.937	0.963	0.873	0.619
P5O20	0.998	0.990	0.994	0.967	0.788	P6O40	0.990	0.906	0.946	0.854	0.603
P5O25	0.997	0.984	0.990	0.948	0.729	P6O45	0.989	0.874	0.928	0.831	0.590
P5O30	0.996	0.969	0.982	0.922	0.708	P6O50	0.988	0.842	0.909	0.812	0.583

Figure 1

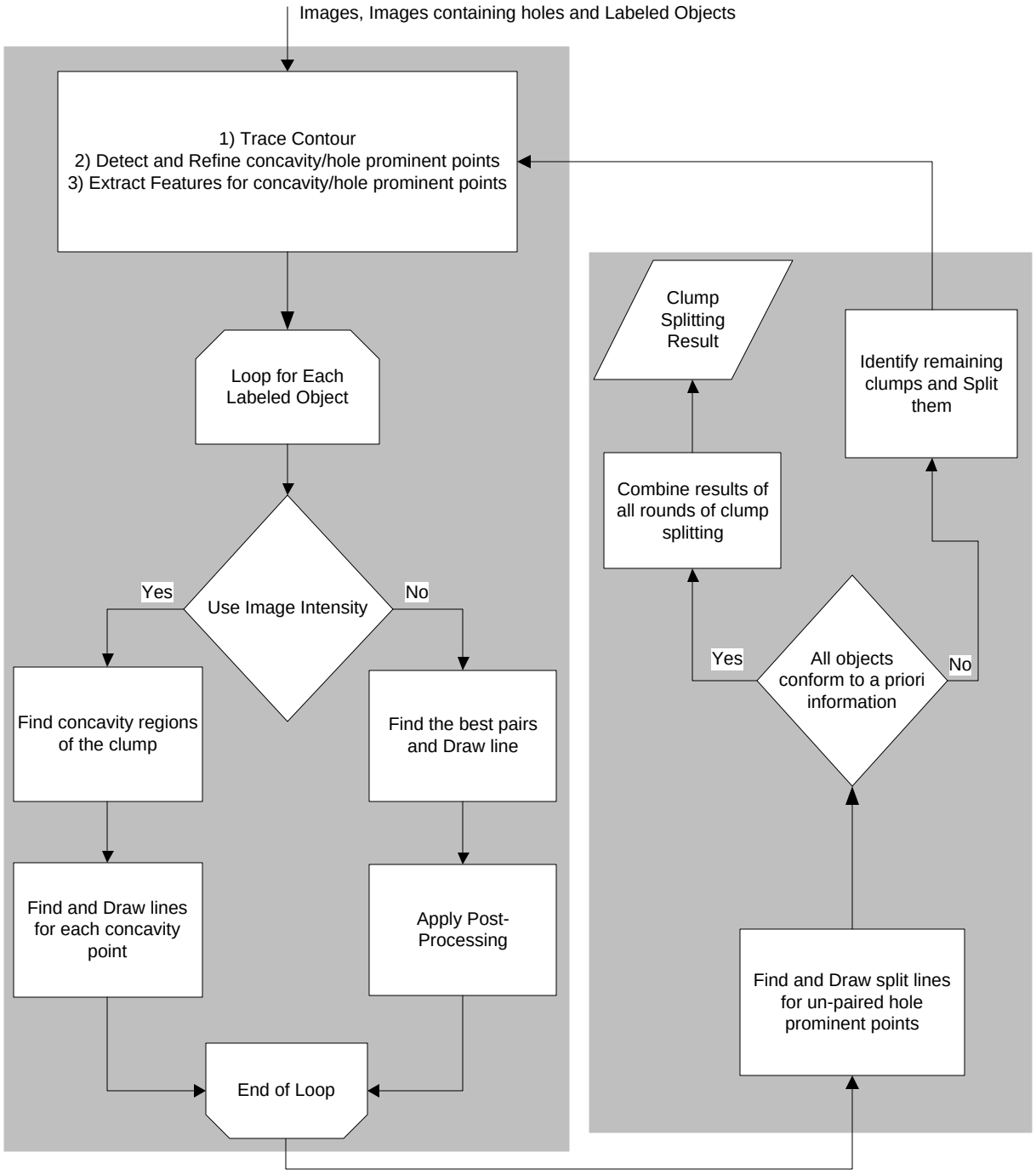

Images, Images containing holes and Labeled Objects
1) Trace Contour
2) Detect and Refine concavity/hole prominent points
3) Extract Features for concavity/hole prominent points
Loop for Each Labeled Object
Use Image Intensity
Yes
No
Find concavity regions of the clump
Find and Draw lines for each concavity point
Find the best pairs and Draw line
Apply Post-Processing
End of Loop
Clump Splitting Result
Identify remaining clumps and Split them
Combine results of all rounds of clump splitting
All objects conform to a priori information
Yes
No
Find and Draw split lines for un-paired hole prominent points

Figure 2a

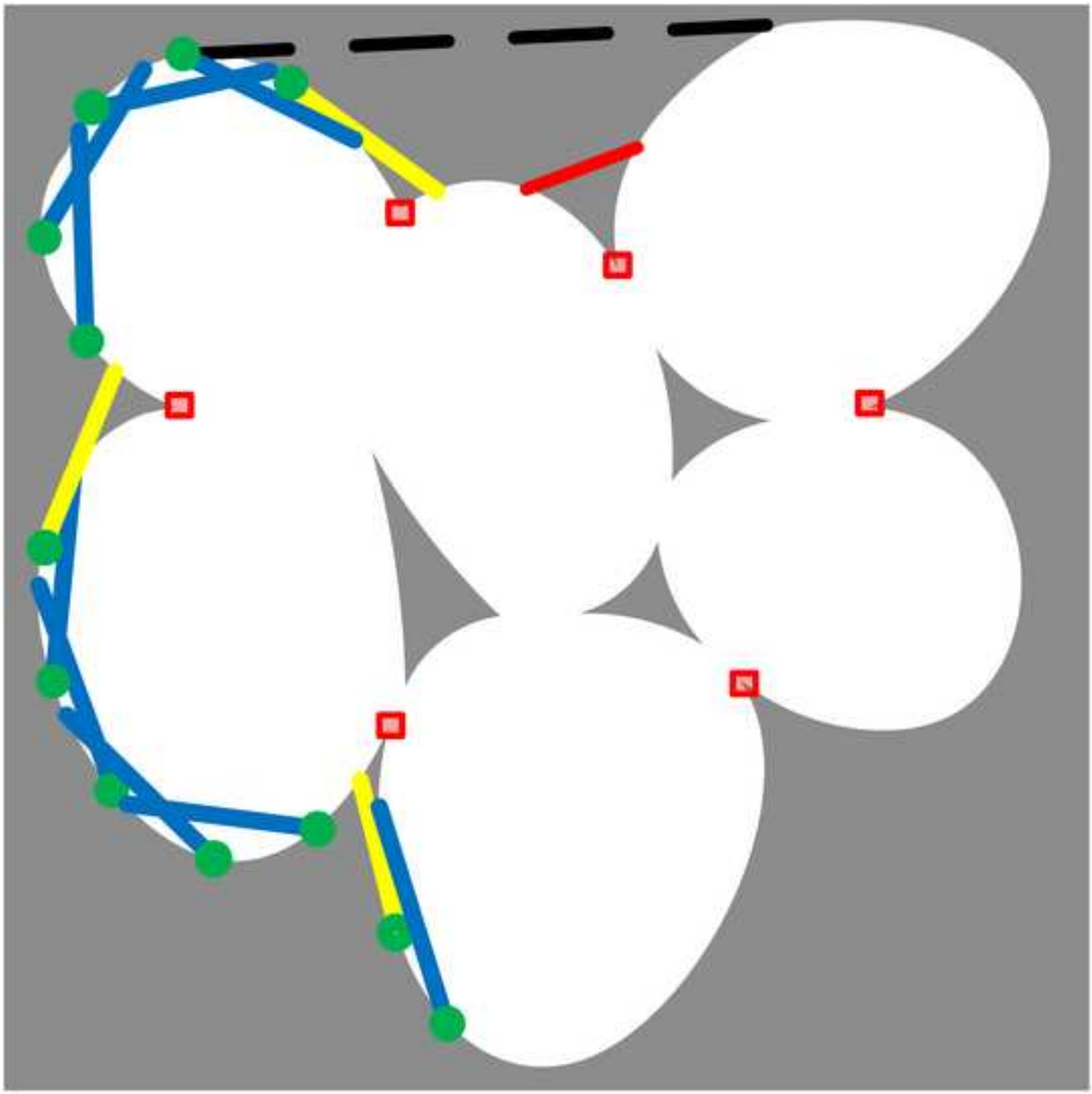


Figure 2b

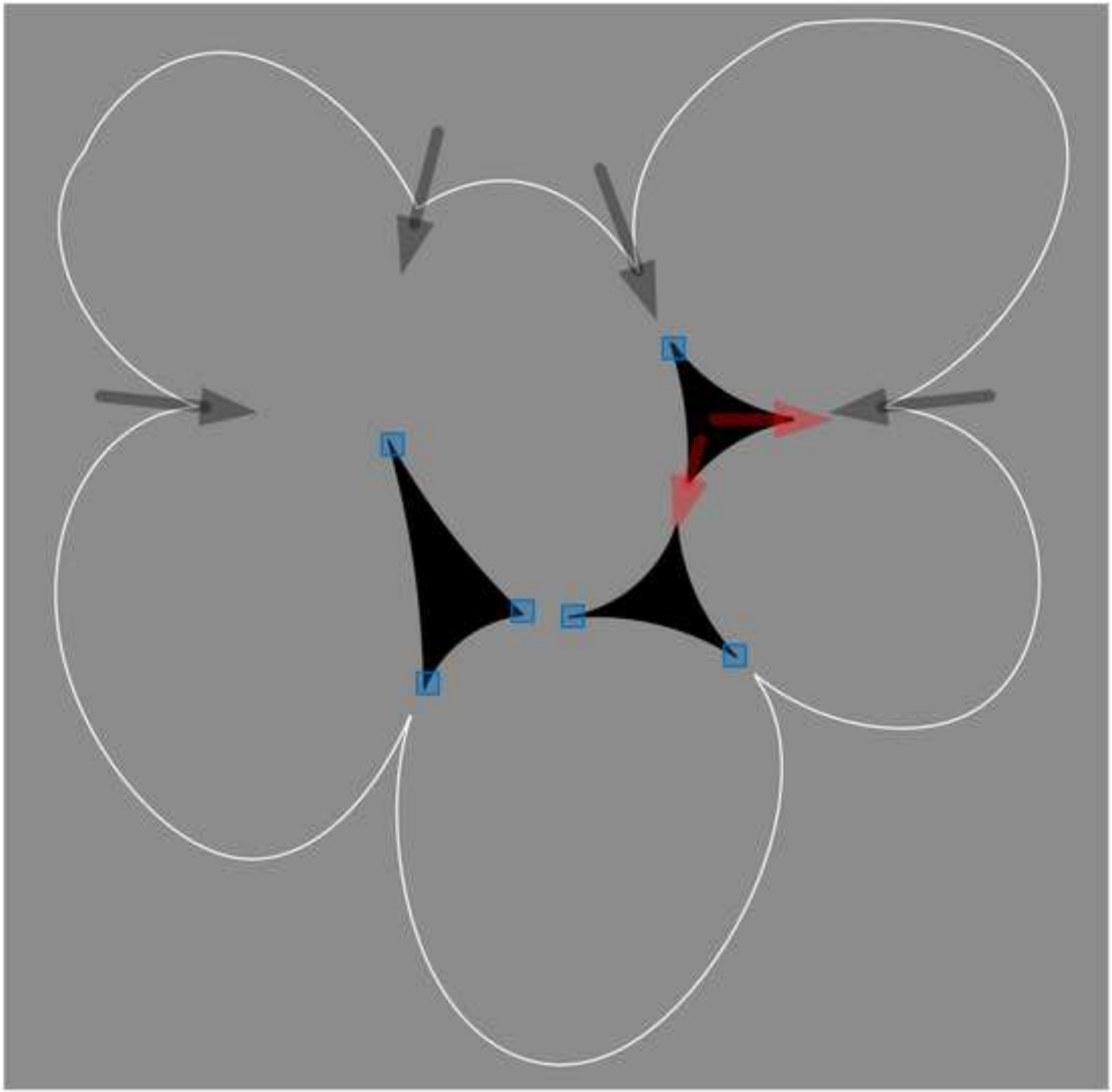


Figure 3a

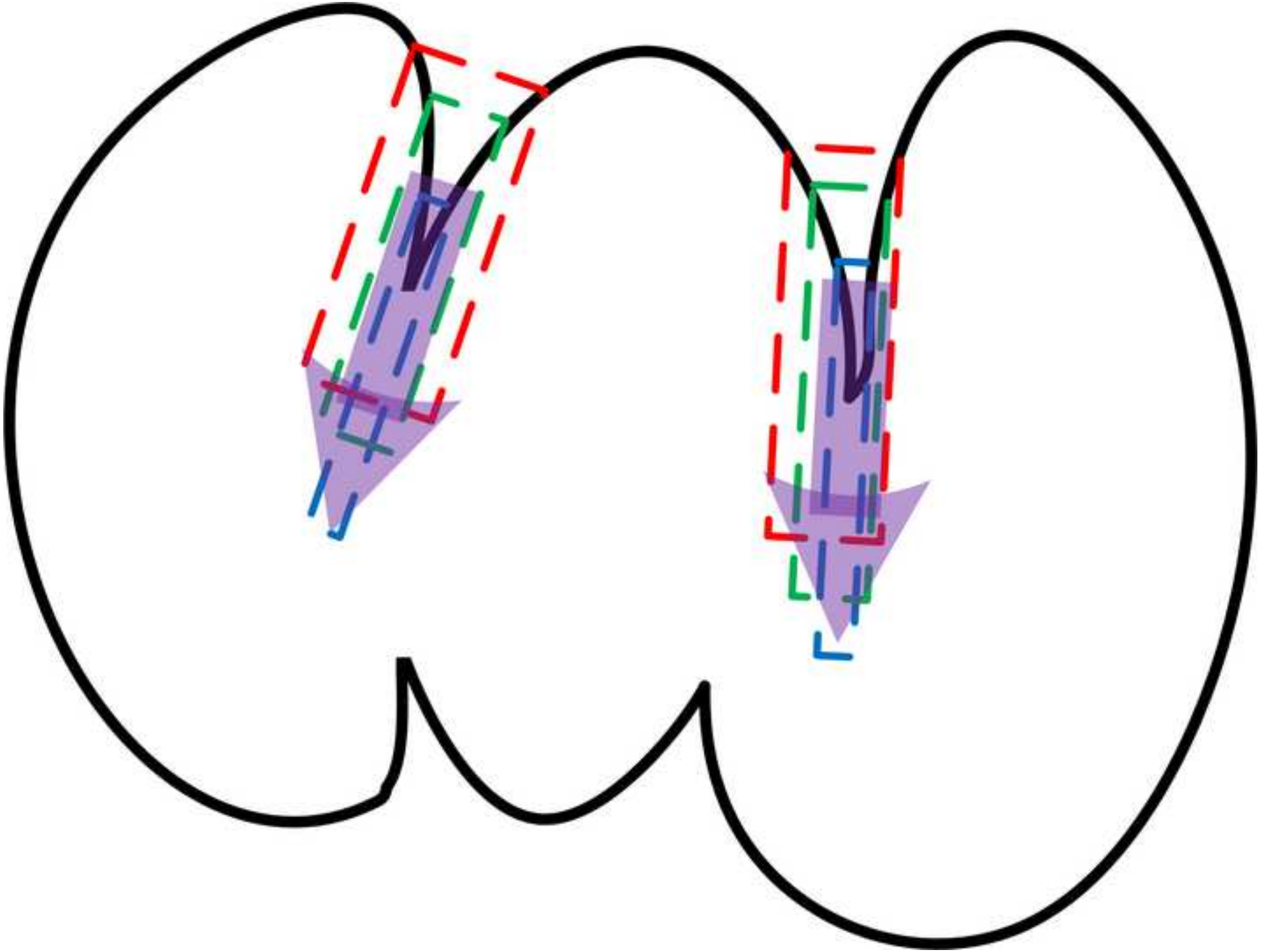


Figure 3b

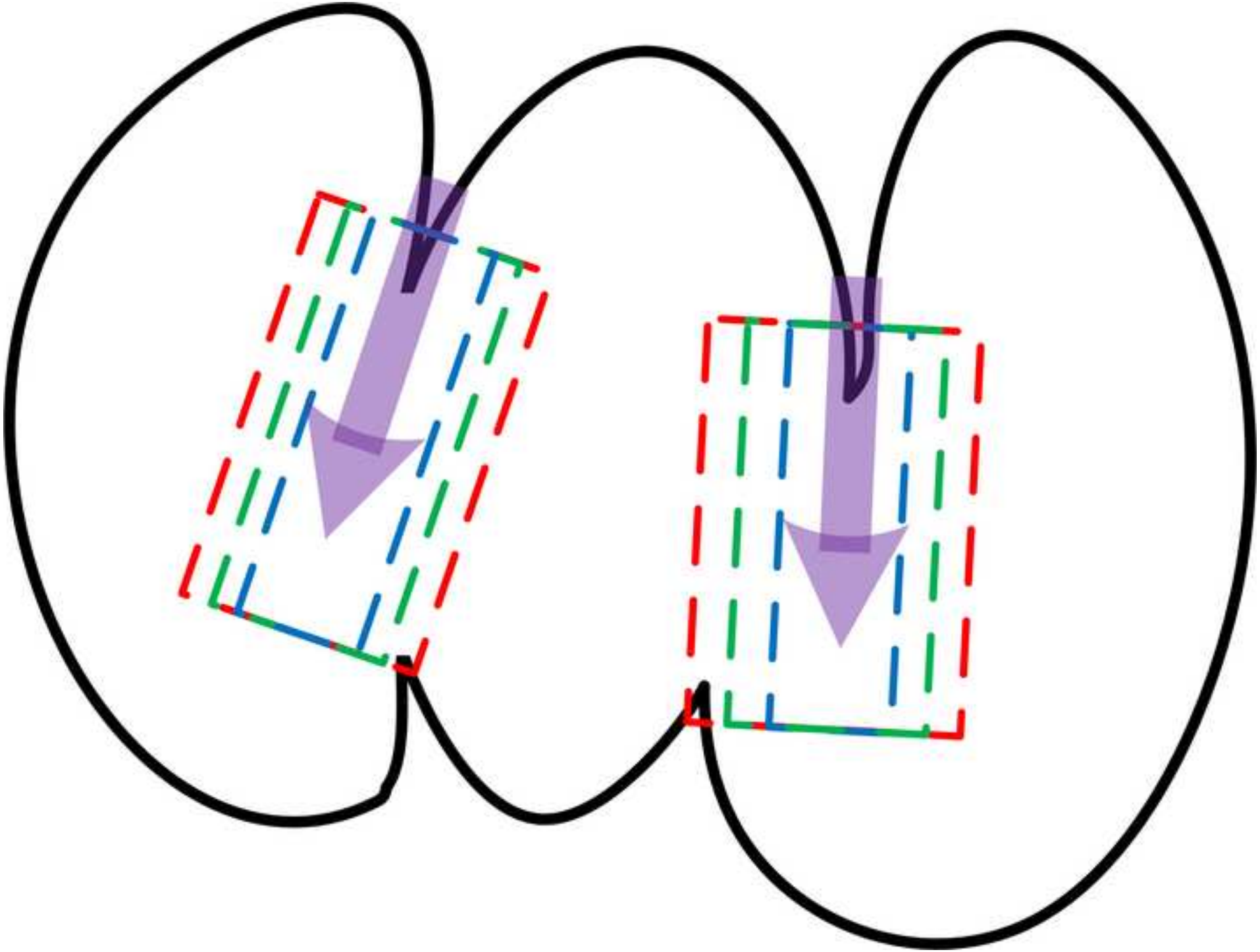


Figure 4a

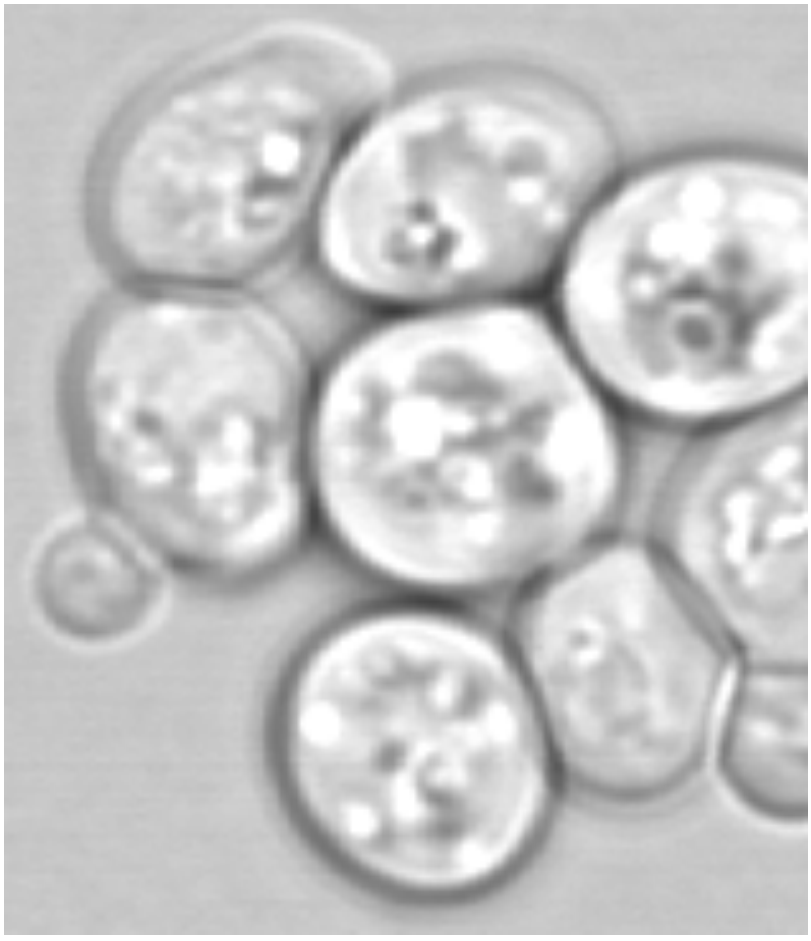


Figure 4b

Figure 4c

Figure 4d

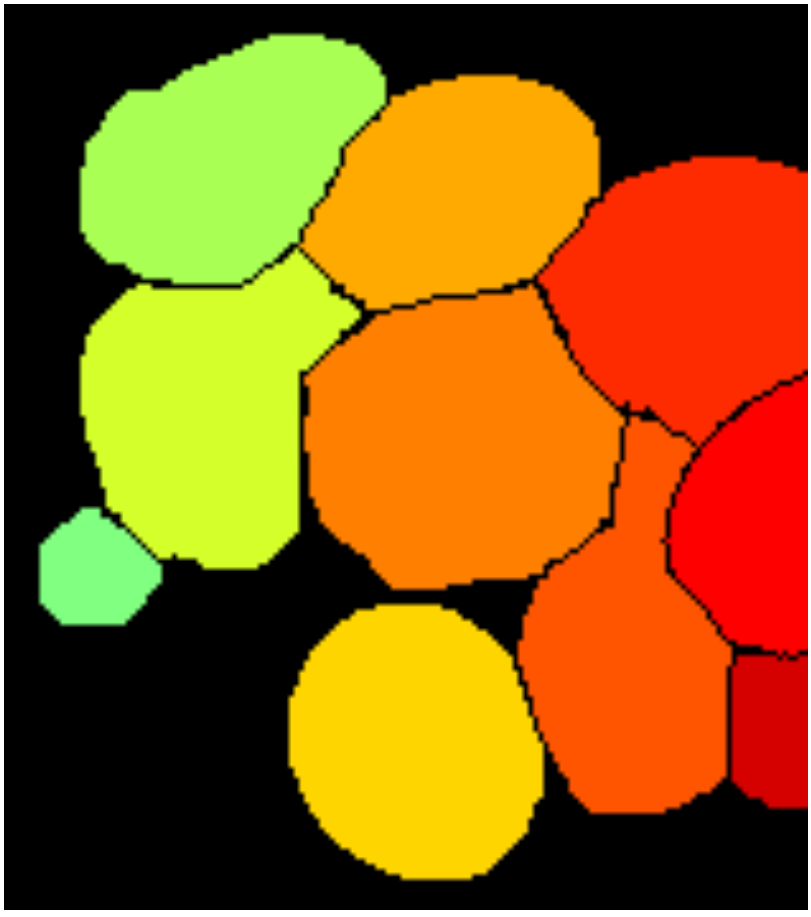


Figure 5a

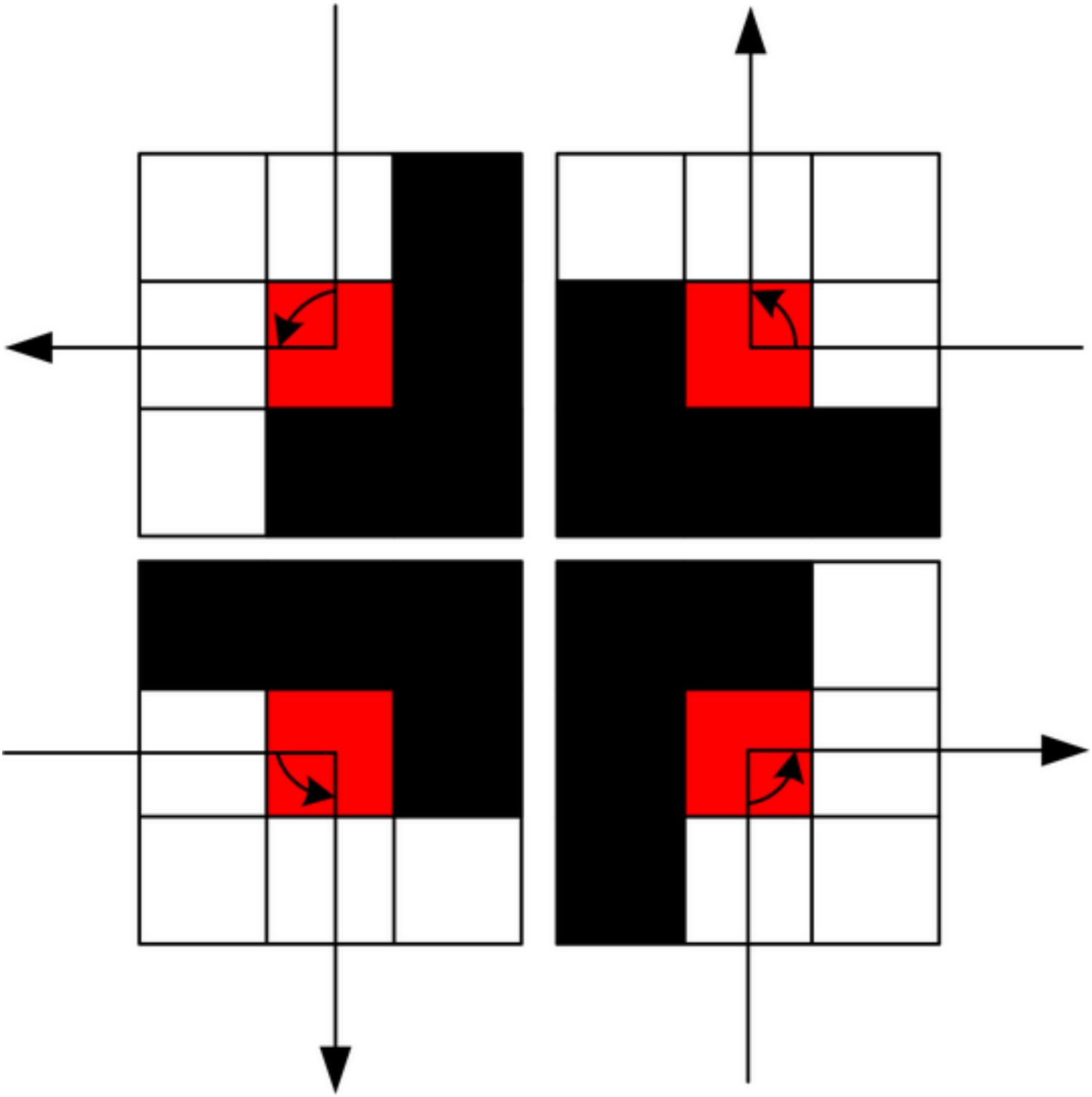


Figure 5b

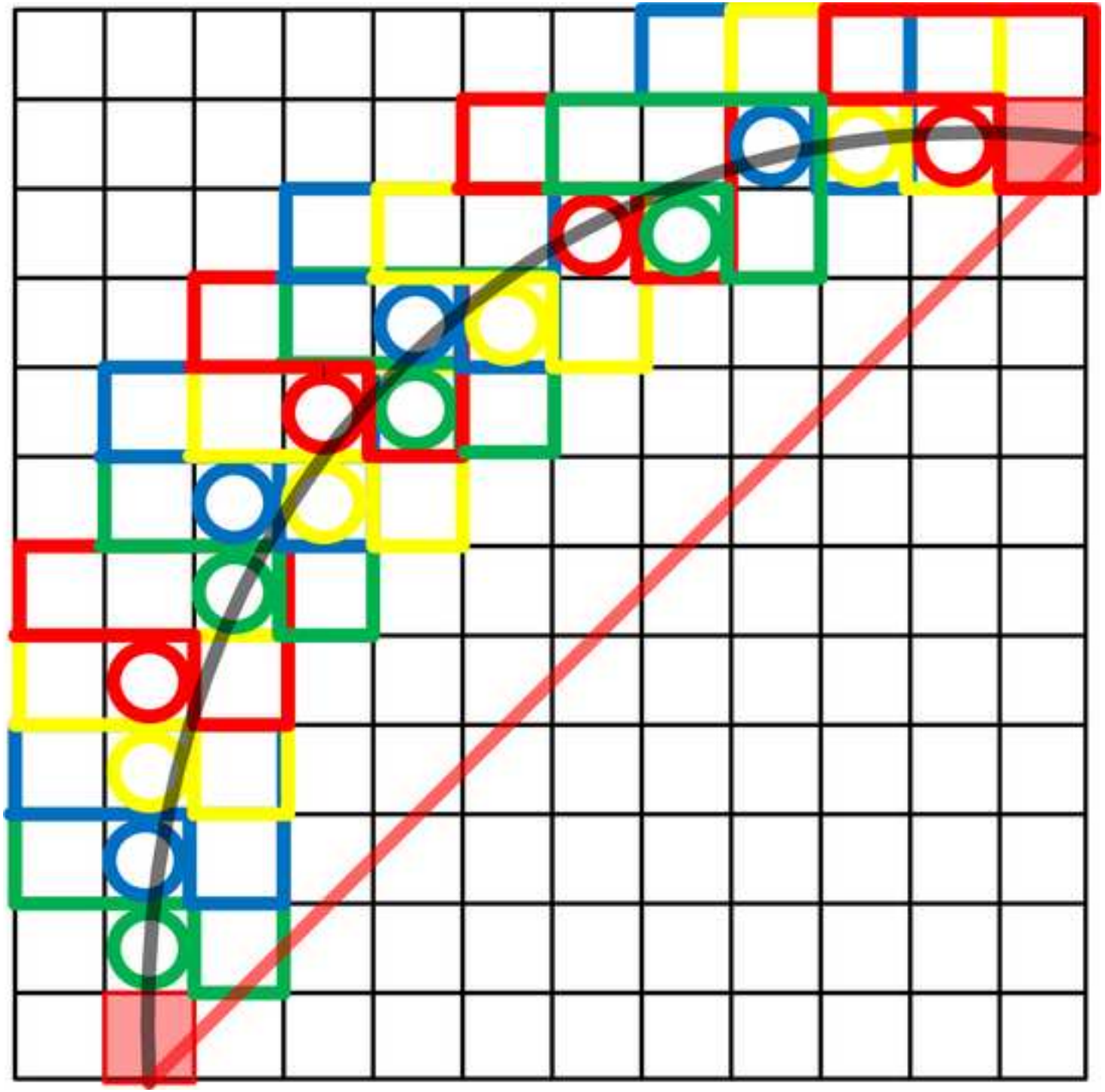


Figure 6a

Figure 6b

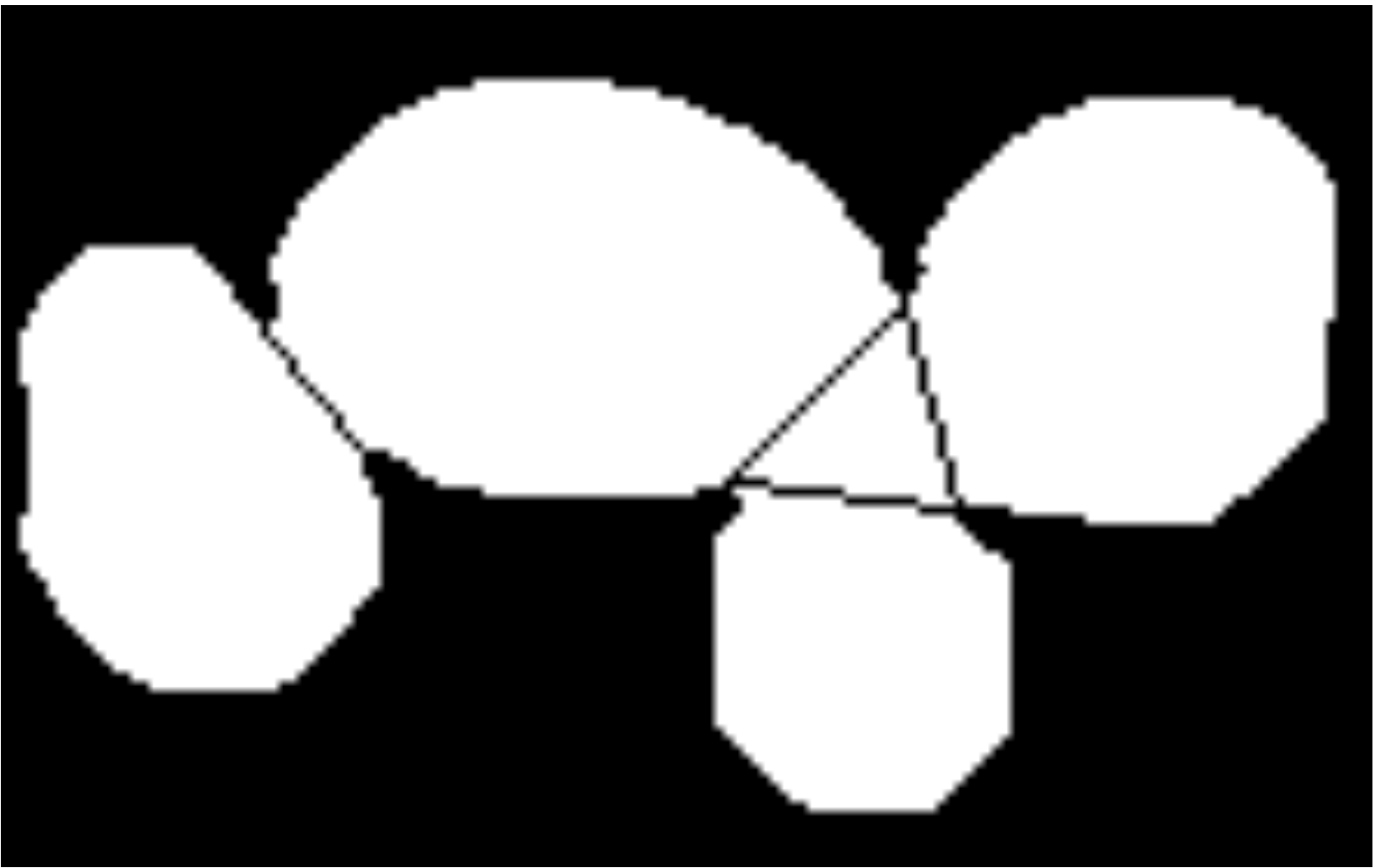


Figure 7a

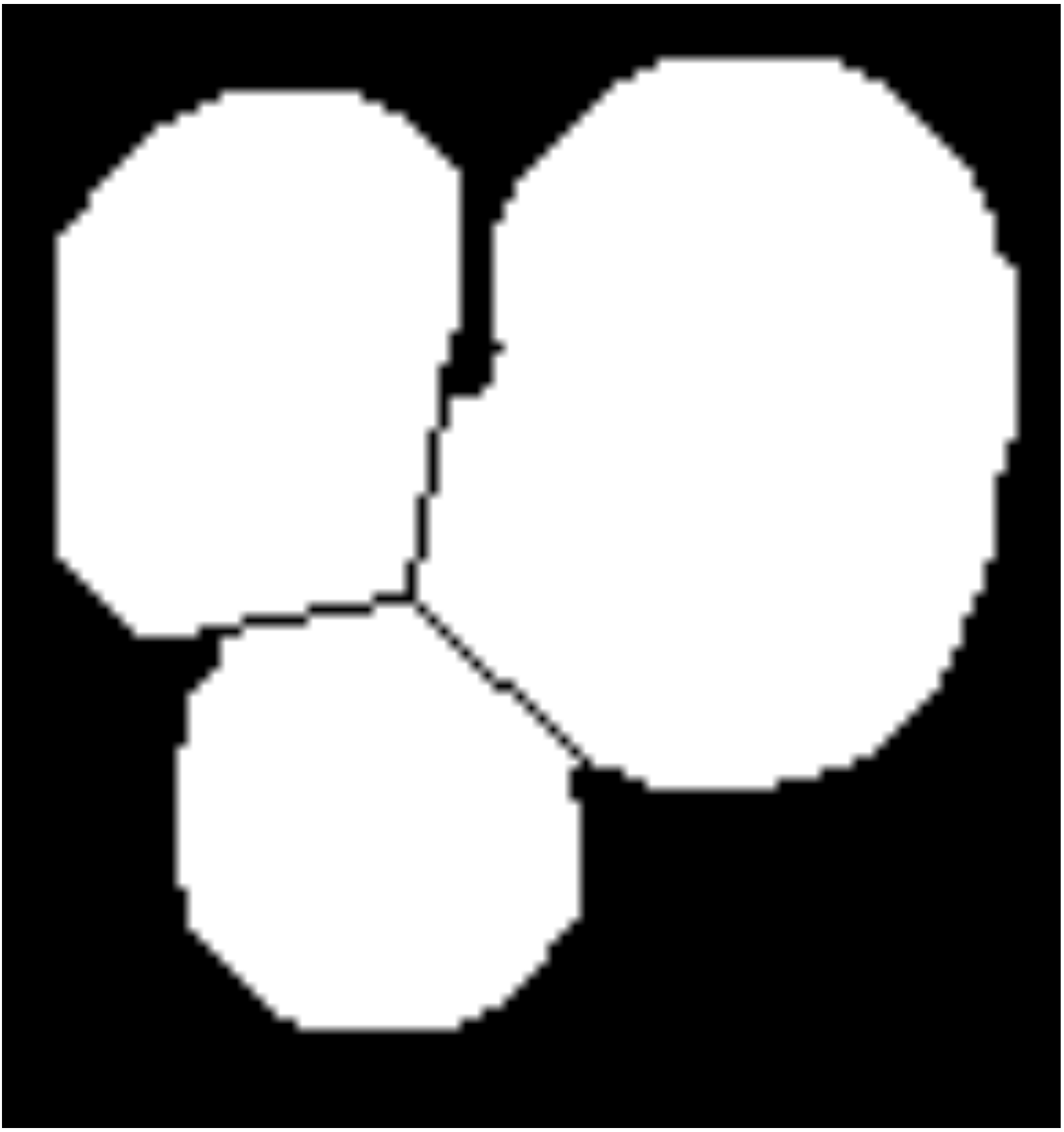


Figure 7b

Figure 8a

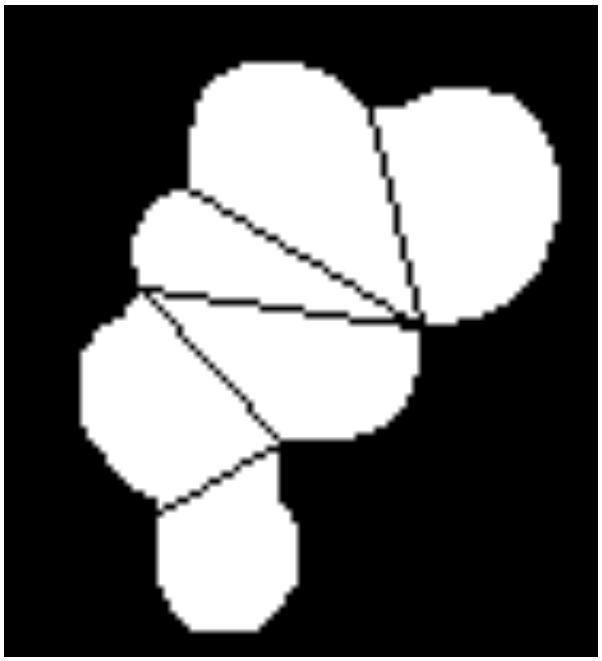


Figure 8b

Figure 9a

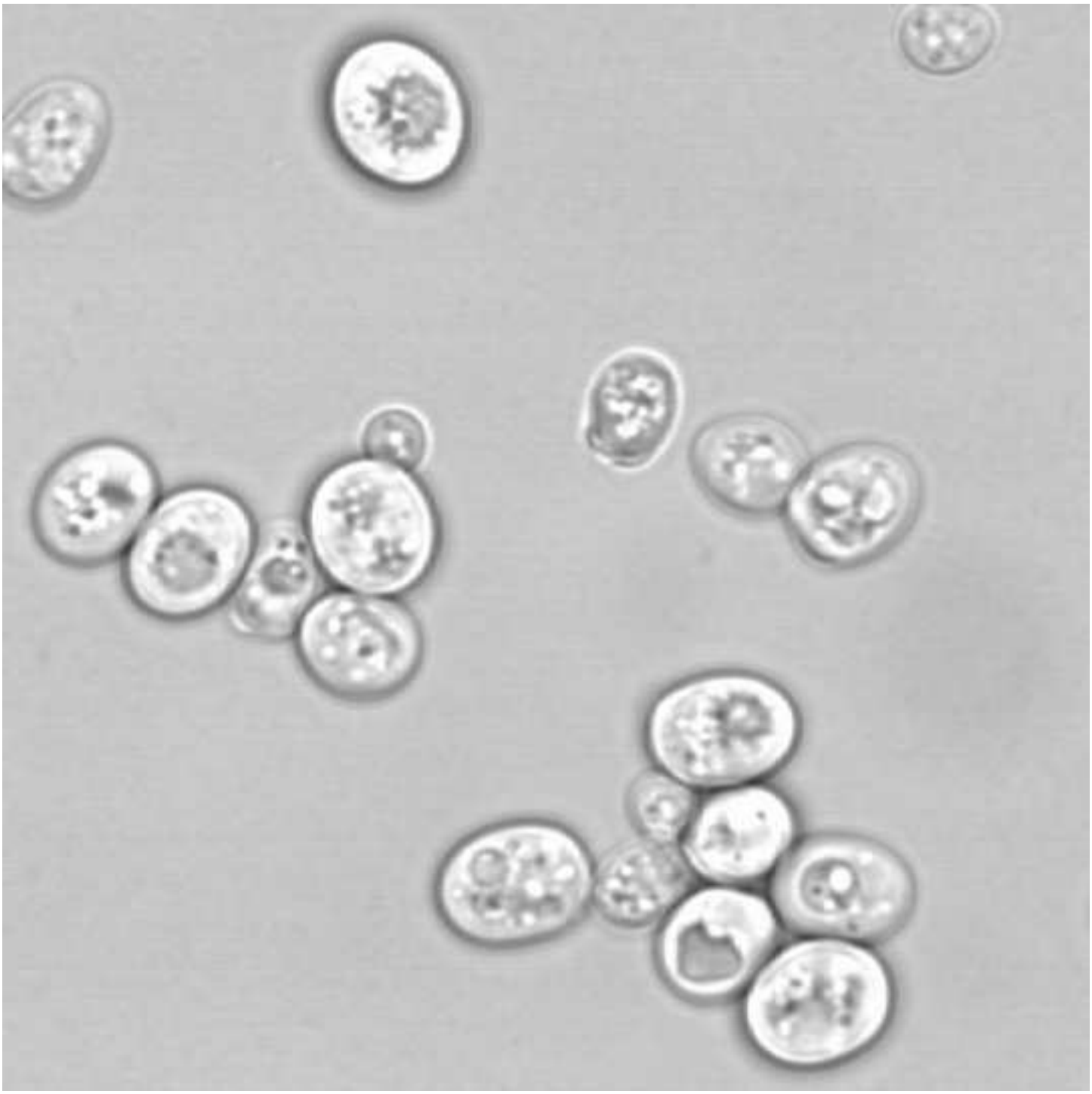


Figure 9b

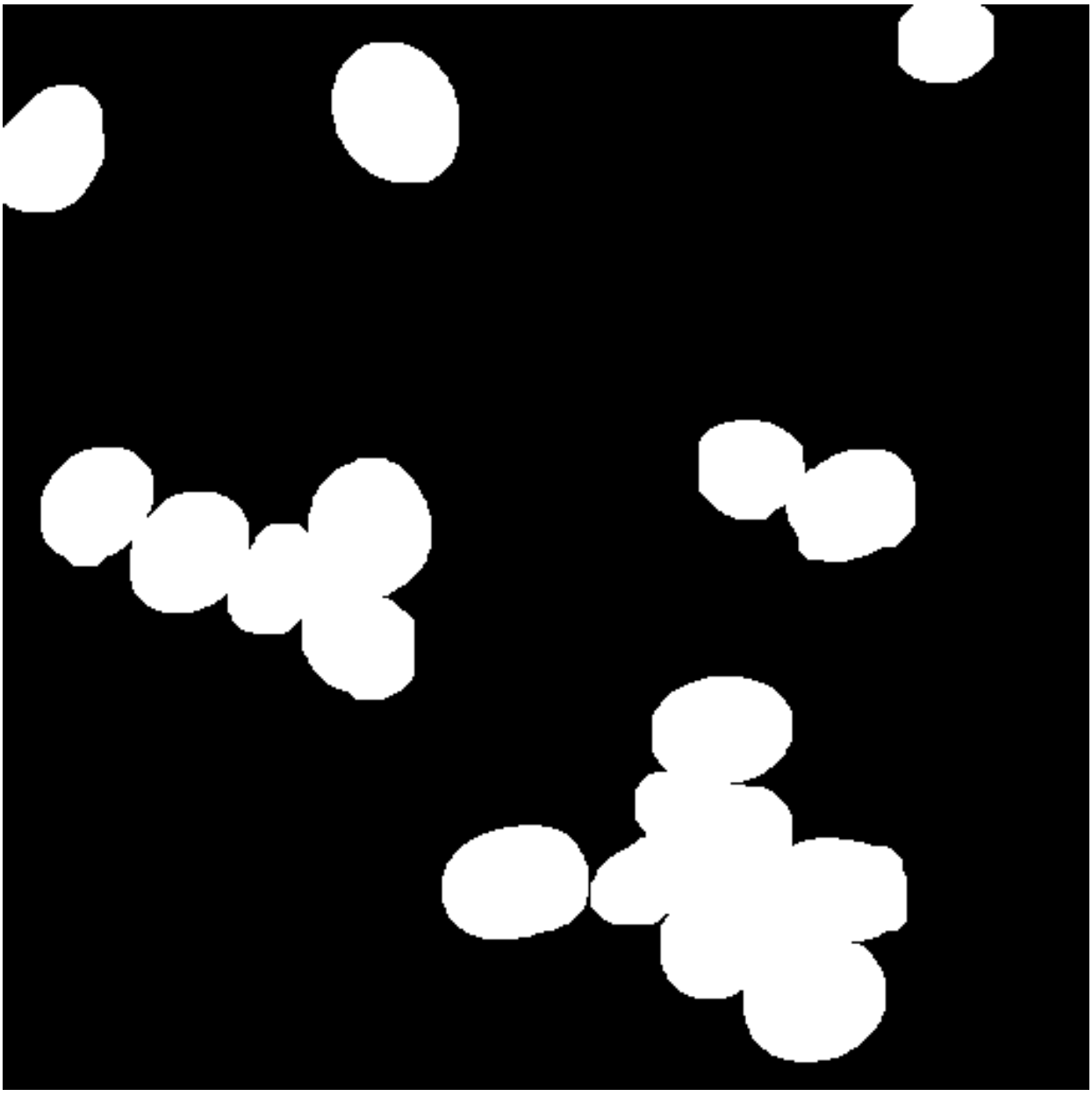


Figure 9c

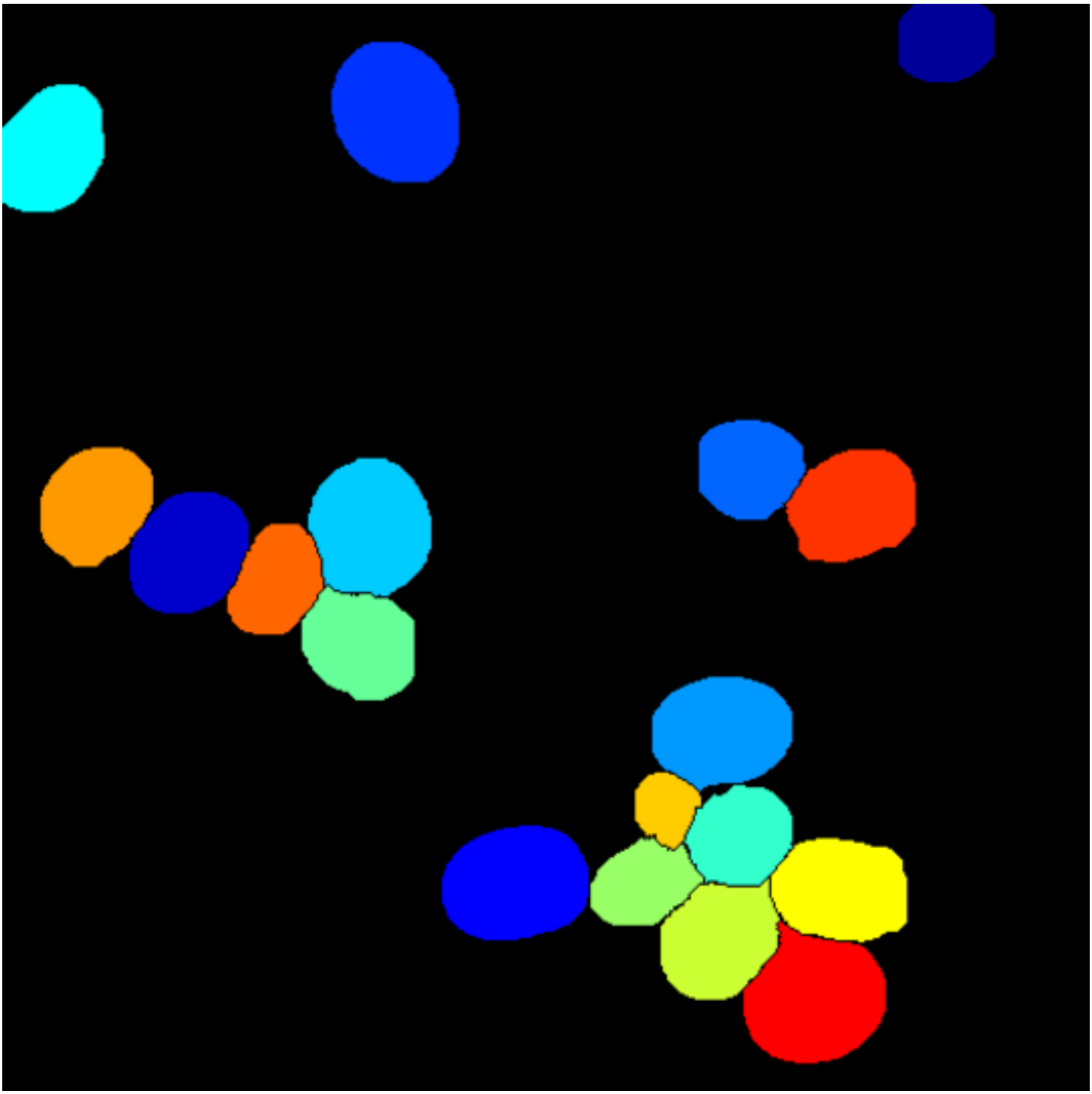


Figure 9d

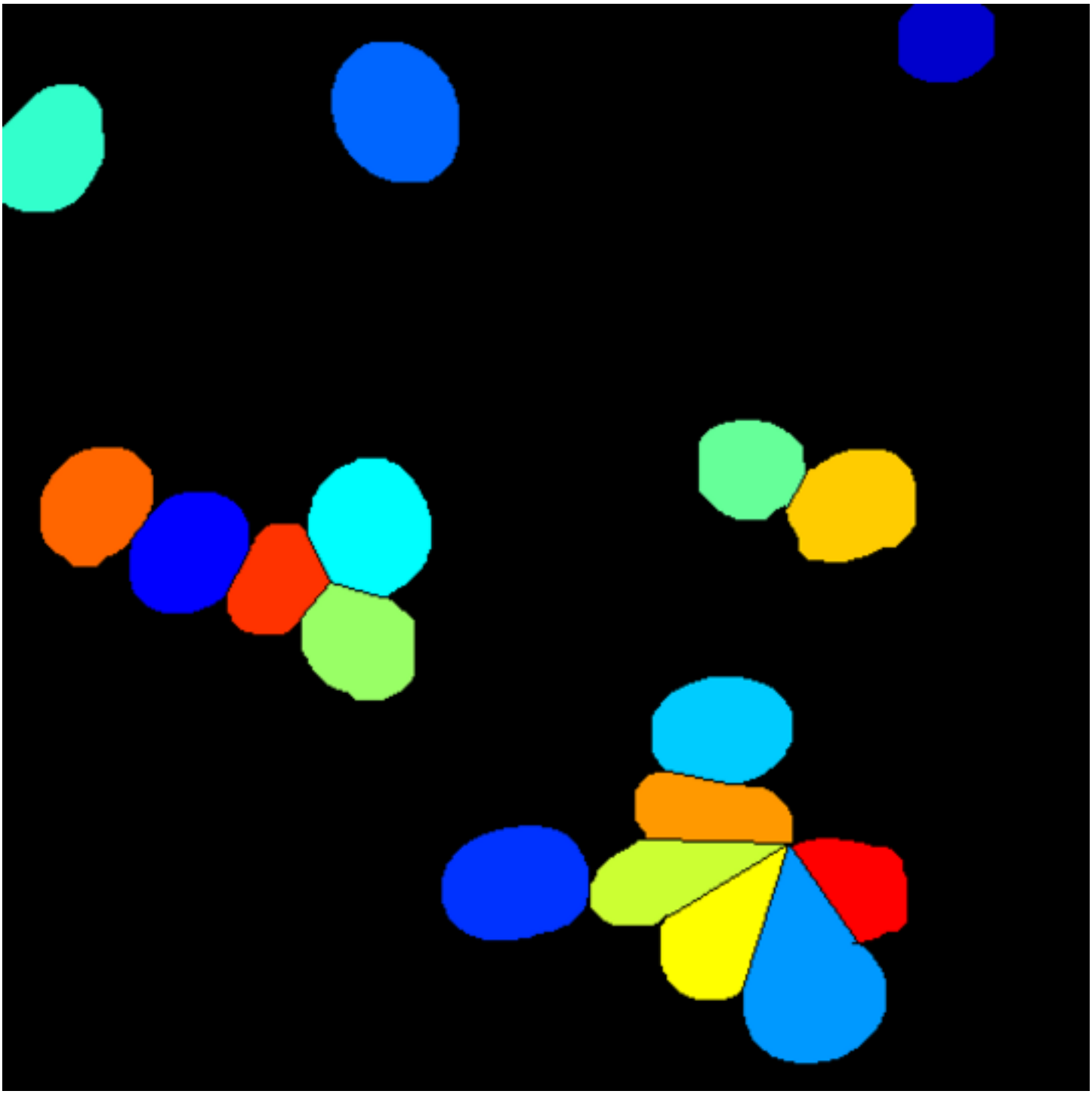


Figure 10a

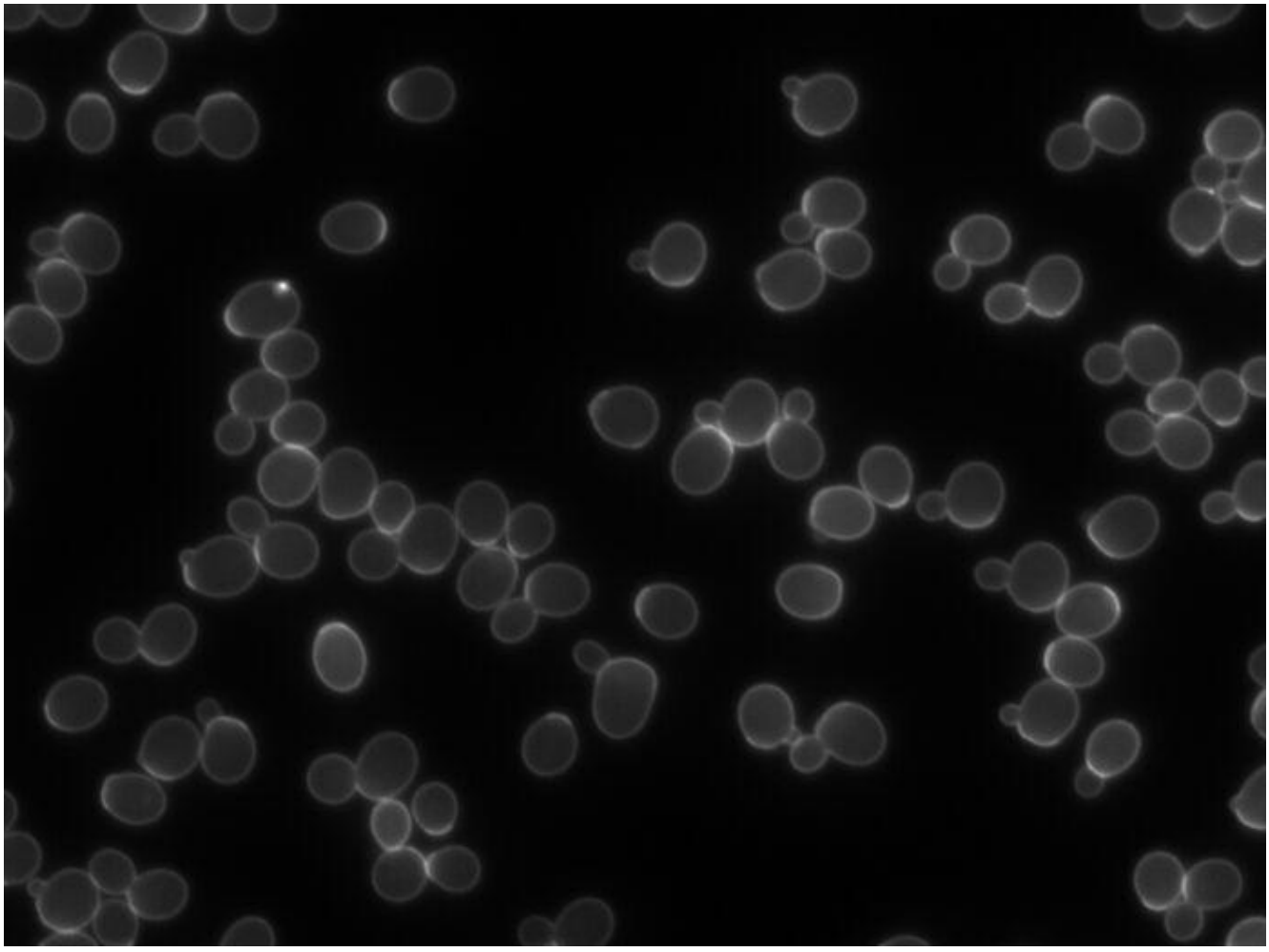


Figure 10b

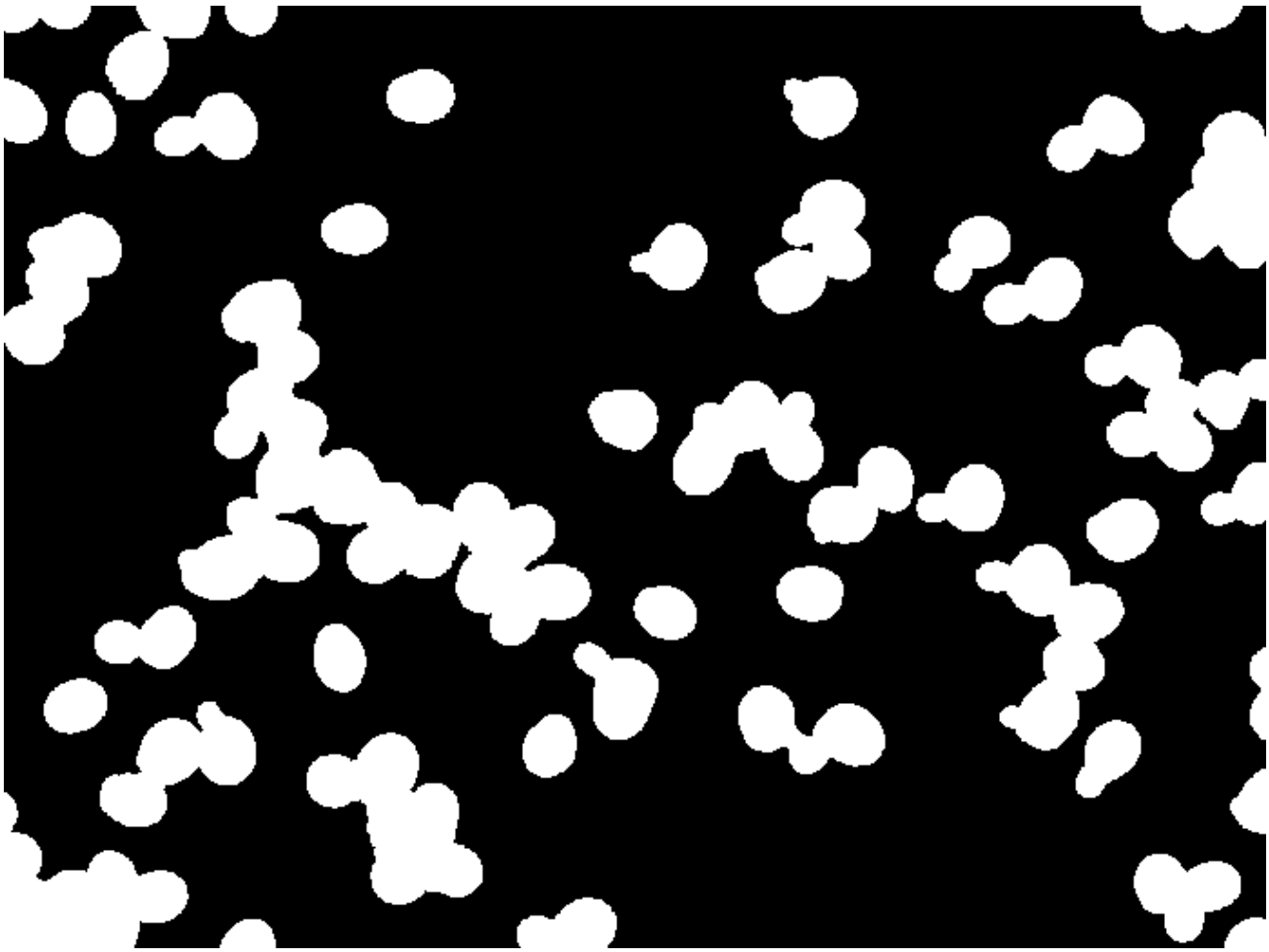


Figure 10c

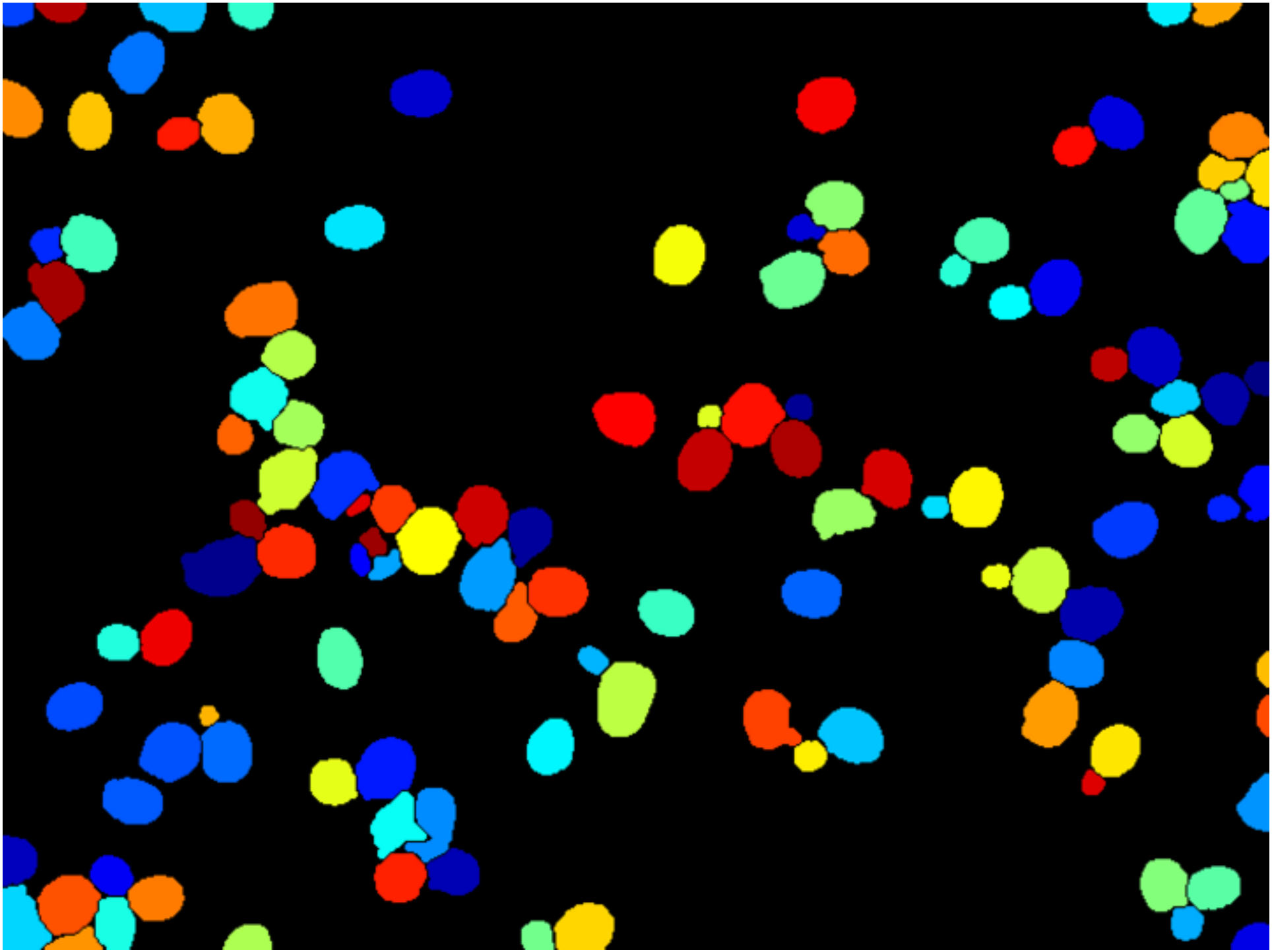


Figure 10d

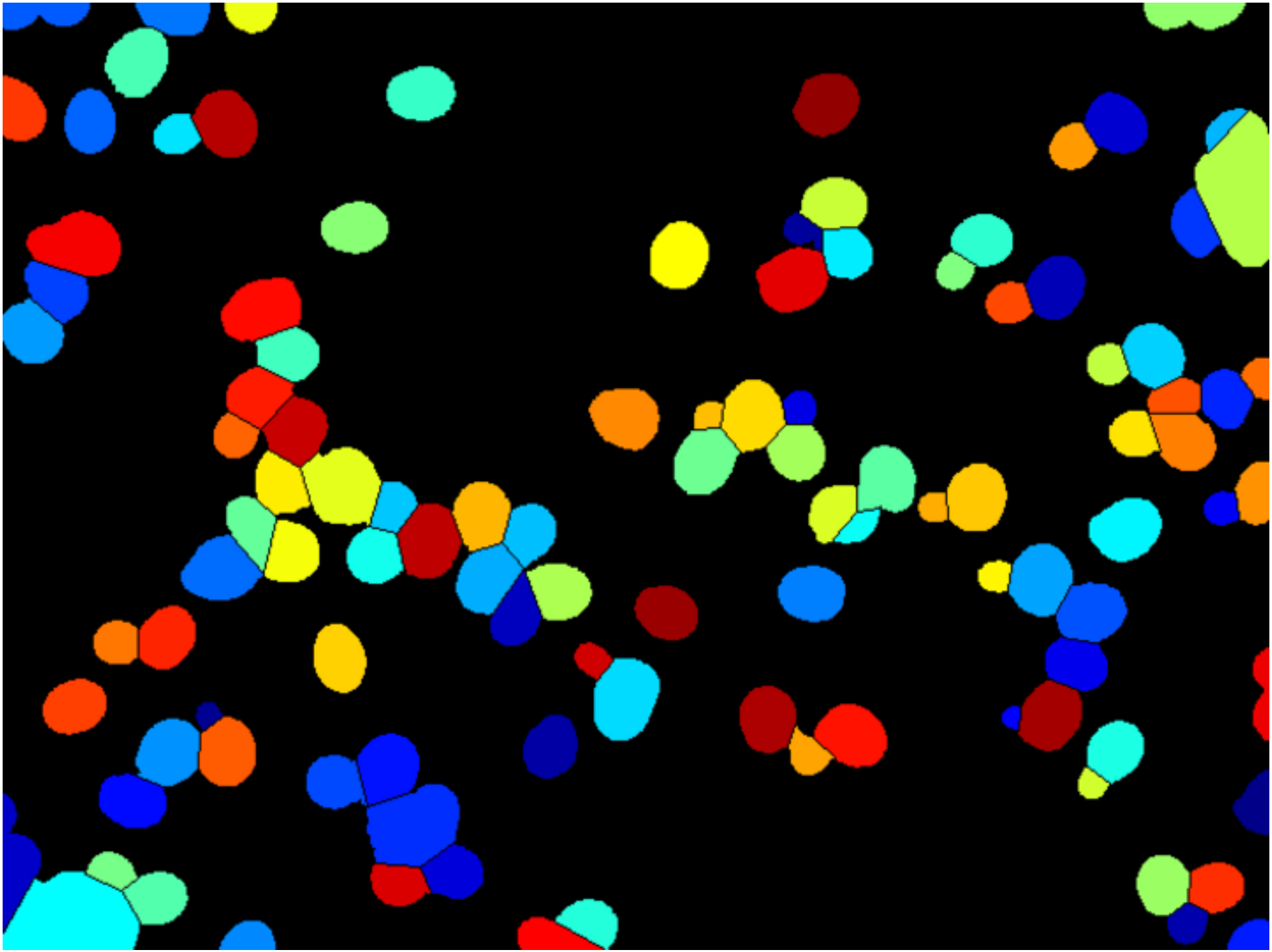


Figure 11a

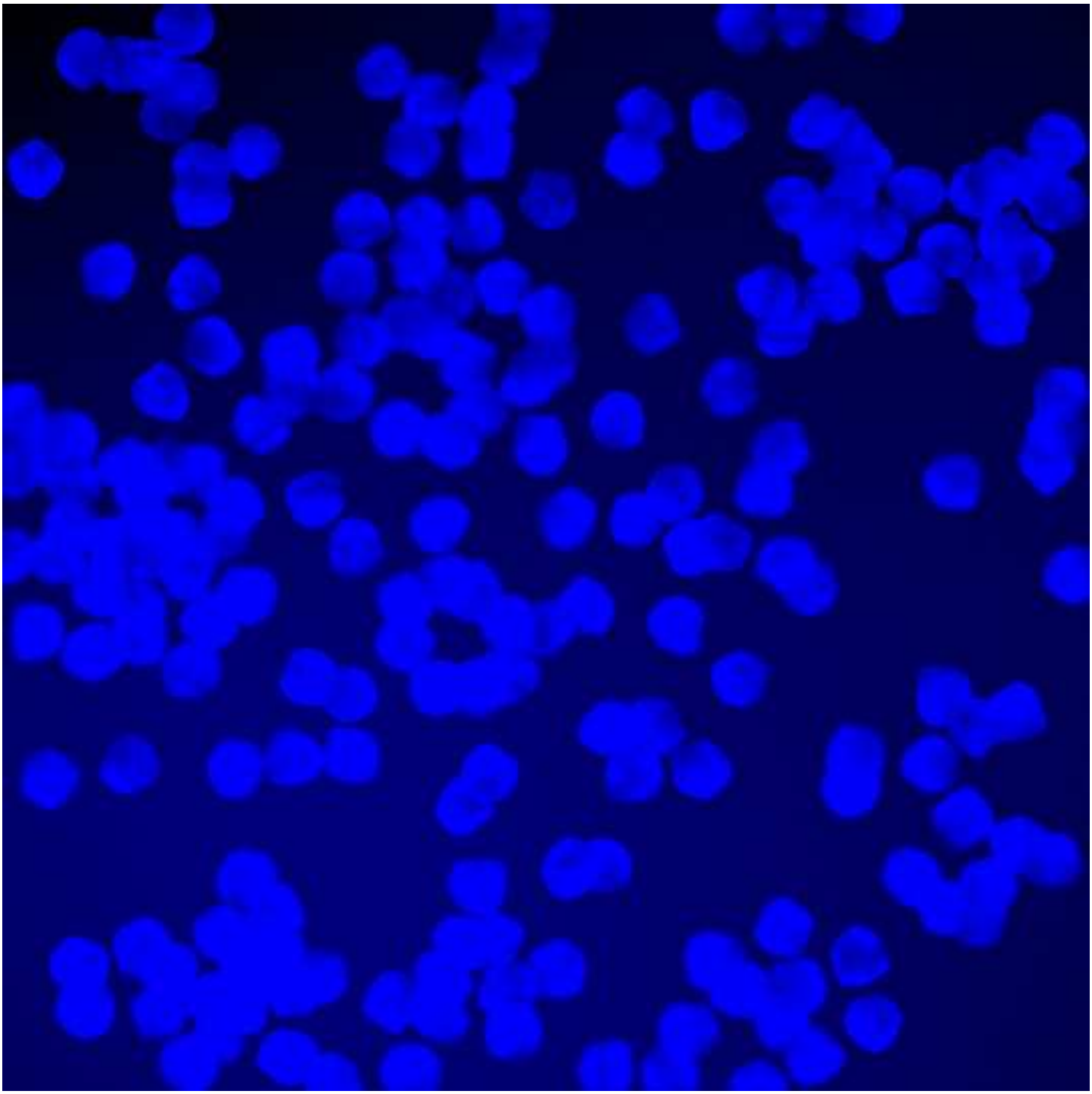


Figure 11b

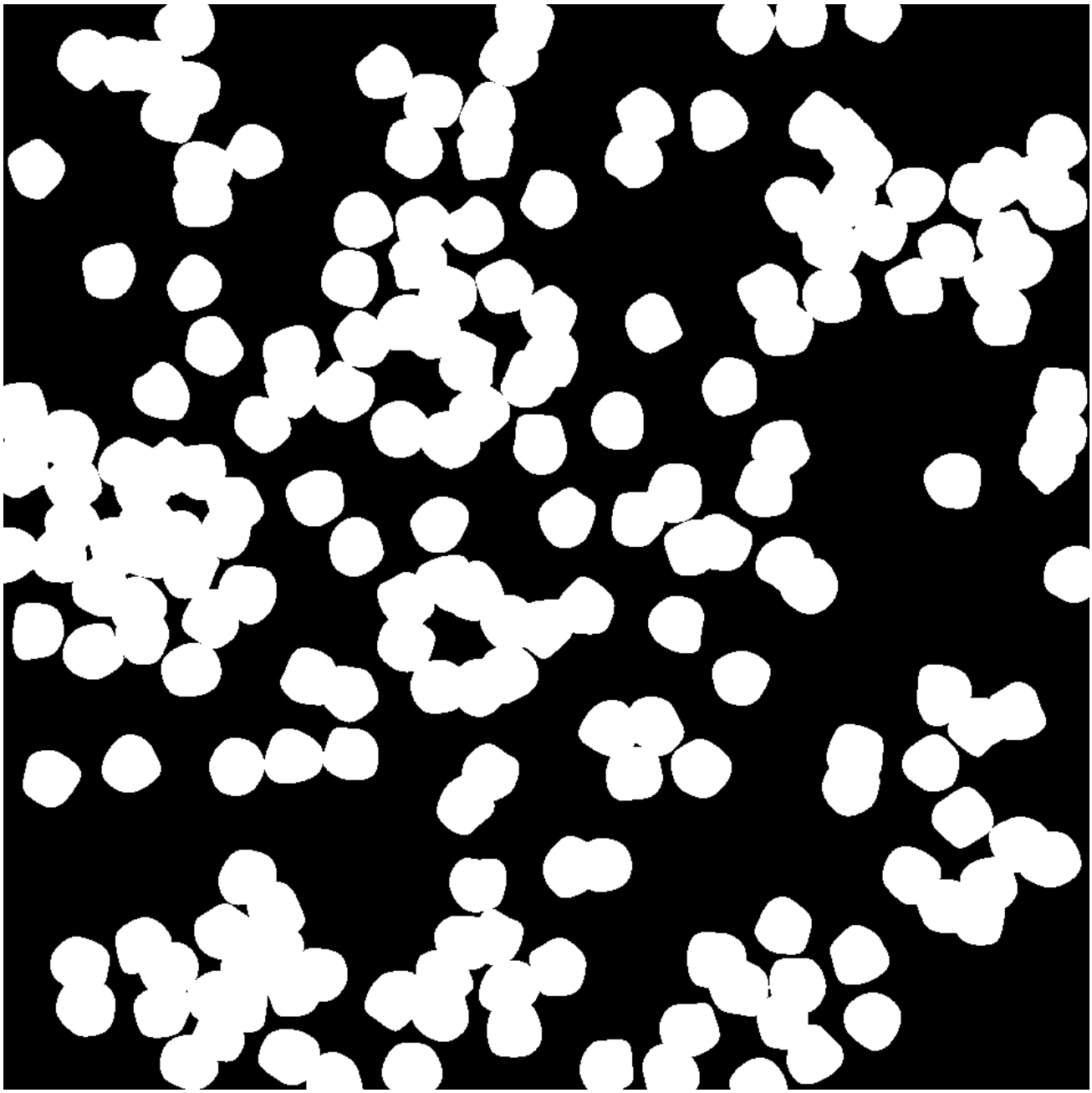


Figure 11c

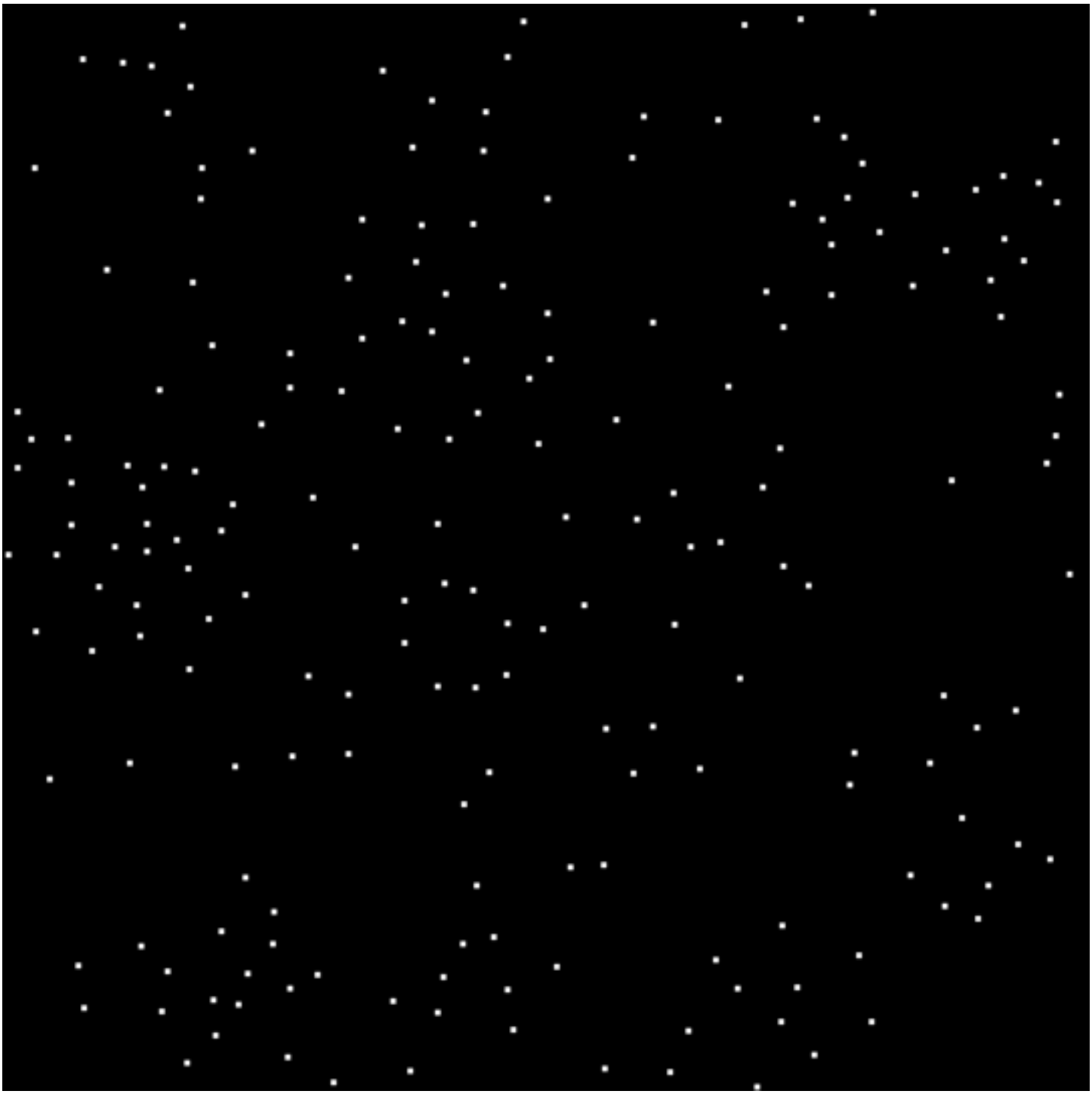
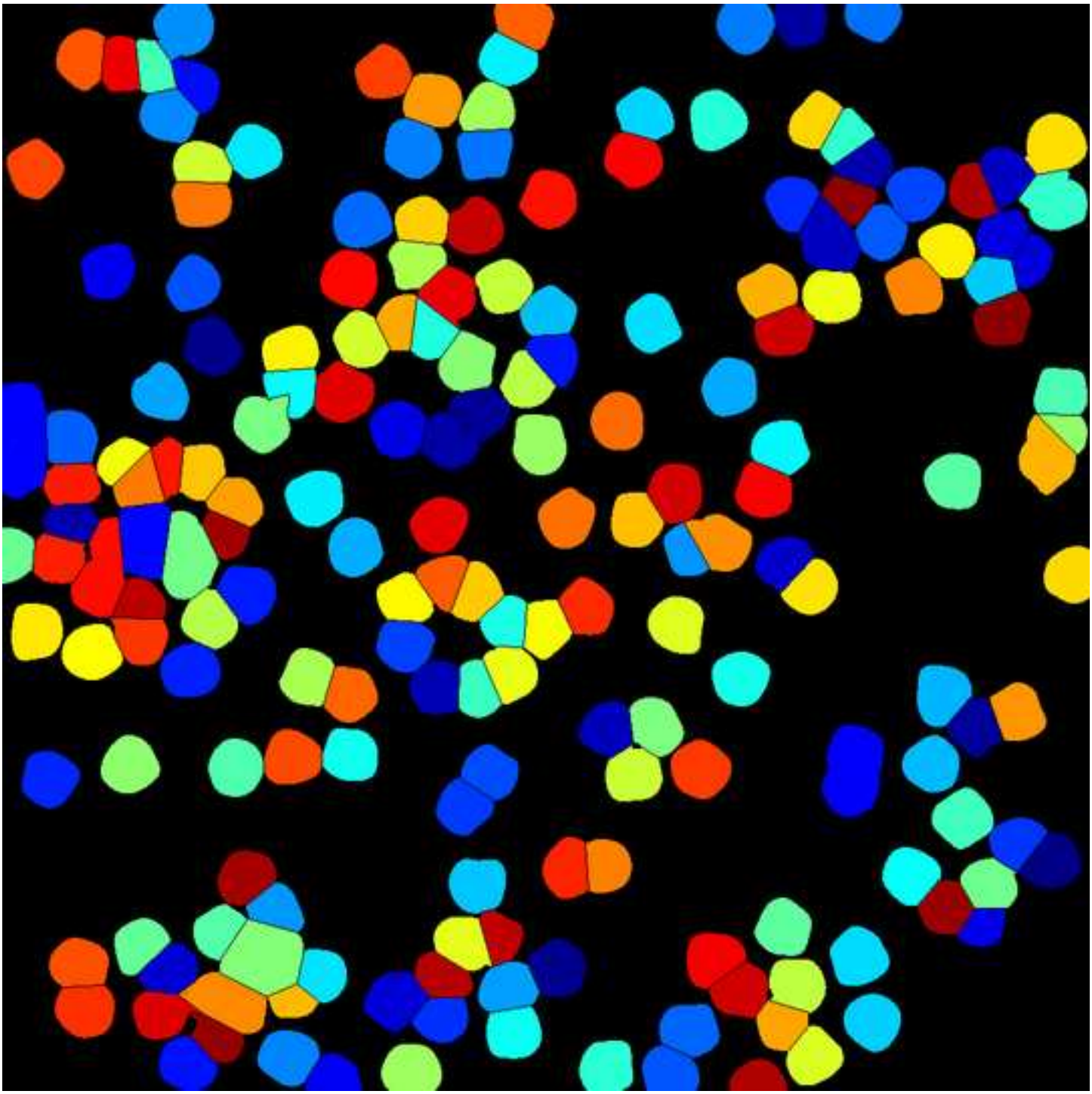


Figure 11d



Publication V

S.S. Hassan*, M. Farhan*, R. Mangayil, H. Huttunen, and T. Aho, "Bioprocess data mining using regularized regression and random forests," *BMC Systems Biology*, 7(Suppl 1):S5, August 2013.
(* corresponds to equal contribution and joint first authors)

Hassan et al. BMC Systems Biology 2013, 7(Suppl 1):S5
http://www.biomedcentral.com/1752-0509/7/S1/S5

BMC Systems Biology

RESEARCH Open Access

Bioprocess data mining using regularized regression and random forests

Syeda Sakira Hassan[1*†], Muhammad Farhan[1†], Rahul Mangayil[2], Heikki Huttunen[1], Tommi Aho[2]

From 10th International Workshop on Computational Systems Biology
Tampere, Finland. 10-12 June 2013

Abstract

Background: In bioprocess development, the needs of data analysis include (1) getting overview to existing data sets, (2) identifying primary control parameters, (3) determining a useful control direction, and (4) planning future experiments. In particular, the integration of multiple data sets causes that these needs cannot be properly addressed by regression models that assume linear input-output relationship or unimodality of the response function. Regularized regression and random forests, on the other hand, have several properties that may appear important in this context. They are capable, e.g., in handling small number of samples with respect to the number of variables, feature selection, and the visualization of response surfaces in order to present the prediction results in an illustrative way.

Results: In this work, the applicability of regularized regression (Lasso) and random forests (RF) in bioprocess data mining was examined, and their performance was benchmarked against multiple linear regression. As an example, we used data from a culture media optimization study for microbial hydrogen production. All the three methods were capable in providing a significant model when the five variables of the culture media optimization were linearly included in modeling. However, multiple linear regression failed when also the multiplications and squares of the variables were included in modeling. In this case, the modeling was still successful with Lasso (correlation between the observed and predicted yield was 0.69) and RF (0.91).

Conclusion: We found that both regularized regression and random forests were able to produce feasible models, and the latter was efficient in capturing the non-linearity in the data. In this kind of a data mining task of bioprocess data, both methods outperform multiple linear regression.

Background

Industrial biotechnology exploits processes that use living cells, for instance yeast and various bacteria, to produce products like fine chemicals, active pharmaceutical ingredients, enzymes, and biofuels. The use of living material in manufacturing processes makes the processes challenging to develop and control. Because of the complexity of these tasks, computational modeling and data analysis are used to improve the yield, reproducibility and robustness in bioprocesses. On the other hand, the regulatory demands on pharmaceutical manufacturing processes are increasing and, for example, the United States Food and Drug Administration emphasize the importance of model-aided process development in its process analytical technology (PAT) initiative [1]. One of the important steps in process development is maximizing the product yield. In practice, the process optimization includes (1) identifying the process parameters that have most impact to the product yield and, (2) determining their optimal values. This data analysis task includes few features that are specific to the application area. For example, the number of process parameters (predictors) may be large with respect to the number of samples, the predictors may contain either numerical or categorical values, the datasets may contain

* Correspondence: sakira.hassan@tut.fi
† Contributed equally
[1]Department of Signal Processing, Tampere University of Technology, Tampere, P.O. Box 553, 33101, Finland
Full list of author information is available at the end of the article

missing values and, finally, the relationship among the predictors and product yield may be non-linear.

To build a model for data analysis requires selection of important features while leaving out the rest. Several feature selection methods have been proposed but the results tend to vary, as generalization of the solution is problematic. Typical issues are data redundancy, outliers and feature dependencies [2,3].

Methods

In this work, we have used three alternative approaches to model bioprocess data: multiple linear regression, regularized regression and random forests. The analyses were performed using MATLAB [4] and RF-ACE tool [5].

Multiple linear regression

In multiple linear regression, the response variable is modeled as a linear combination of multiple predictor variables. The general model can be expressed as

$$y = \beta_0 + a_1\beta_1 + a_2\beta_2 + a_3\beta_3 + \ldots + a_p\beta_p \quad (1)$$

where y is the response variable, and a_i and β_i (i = 1, ..., p) are the predictor variables and their coefficients, respectively. The intercept is represented by β_0. Alternatively, Equation (1) can be represented in vector notation by $\mathbf{y} = \mathbf{H}\theta$, where $\mathbf{H}$ is augmented predictor vector given as [$\mathbf{1}\ \boldsymbol{a_1}\ \boldsymbol{a_2}\ \ldots\ \boldsymbol{a_p}$] and θ is the parameter vector.

In spite of being linear with respect to the predictor variables, multiple linear regression models fail to incorporate the underlying non-linear relationships, if it exists, between the predictors and the response variable. However, the model restricts only the coefficients to be linearly related, while the predictor variables can be non-linear. This gives a provision of including additional non-linearly transformed predictor variables in the linear regression modeling. The advantage of using such variables in regression analysis is that the non-linear behavior in data and interaction between different variables are incorporated while the model remains linear and easily interpretable. This is a typical procedure applied in traditional response surface modeling when constructing models with quadratic terms and interactions of terms. Increasing the number of parameters in this way, however, causes high-dimensional predictor vector which results in over-fitting and the loss of generality. Moreover, if the number of samples is small, increasing the parameter vector size by these transformations may cause rank deficiency or multicollinearity of the prediction vector. In such cases, standard regression modeling may either fail, rank deficiency may cause non-invertible matrix thus making parameter estimation difficult, or the estimates it gives for parameter vector are prone to give low prediction accuracy. Hence, regularization is a key process in solving such cases. It produces a sparse parameter vector and also shrinks the coefficients towards zero as well as towards each other [6].

Regularized regression

The research on sparse and regularized solutions has gained increasing interest during the last ten years [7]. This is partly due to advances in measurement technologies, e.g., in molecular biology, where high-throughput technologies allow simultaneous measurement of tens of thousands of variables. However, the measurements are expensive, so typically the number of data points is small. In the field of bioprocess development, the number of variables is not that large but yet enough to hinder the use of many standard data analysis methods. Conventional regression and classification methods are unable to process data with more predictor variables than samples (so called $p >> N$ problem). Regularization methods help in defining a unique solution in this ill-posed problem. These methods shrink some of the coefficients to zero. This not only helps in feature selection but also decreases the variance at the cost of a small increase in bias. However, this has the effect of improving the generalization of the estimate.

In regularized regression, a penalty on the size of the coefficients is added to the error function. Least absolute shrinkage and selection operator (LASSO) [3] is one such technique which uses the L_1 norm of the coefficients as the penalty term to produce *sparse* solutions, i.e., prediction models with several coefficients equal to zero. Since variables with zero coefficients are not used, this procedure essentially acts as an embedded feature selection.

From the description of Equation (1), the L_1 penalized coefficient vector for our linear model is defined as

$$\hat{\boldsymbol{\theta}} = ||\boldsymbol{y} - \boldsymbol{H\theta}||_2^2 + \lambda||\boldsymbol{\theta}||_1 \quad (2)$$

where lambda (λ) is the regularization parameter, $||\theta||_1$ is the L_1-norm of the parameter vector. There exist efficient algorithms for finding solutions for different values of regularization parameters [3].

The result of the regularized regression is quite sensitive to the selection of the parameter λ. In order to appropriately assess the performance, the selection has to be done based on data. The usual approach is to estimate the performance with different λ using a cross-validation approach. Since we also use cross-validation for estimating the performance of the overall method (including the algorithm for selecting λ), this results in two nested cross-validation loops, one for model selection and one for error estimation. More specifically, the outer loop is used for estimating the performance for new data, while the inner loop is used for selection of λ.

Random forests

Decision trees have been studied for decades as a model for various prediction problems. The tree can be either a classification tree or a regression tree, and a common term including both is classification and regression tree (CART). A decision tree is a hierarchical structure, which decides the class (in classification) or the predicted output (regression) by hierarchically comparing feature values with a selected threshold, thus producing a hierarchy of if-then rules. Such combination of rules is most conveniently expressed as a tree, where each input feature comparison corresponds to a node in the tree. Eventually, the leaves of the tree describe the actual output value.

The decision trees can be learned from the data, and the usual approach is to add nodes using a top-down greedy algorithm. In essence, this means dividing the search space into rectangular regions according to the splitting points. The drawback of decision tree is that they are very prone to overlearning. This is one reason why regression trees have later been extended to random forests [8], whose prediction is obtained by averaging the outputs of a large number of regression trees. Due to averaging, random forests are tolerant to overlearning, a typical phenomenon in high-dimensional settings with small sample size, and have thus gained popularity in classification and regression tasks especially in the area of bioinformatics.

In our experiments, we use the RF-ACE implementation in [5]. This implementation is very fast and it takes advantage of the Random Forest with Artificial Ensembles (RF-ACE) algorithm, which enables both feature ranking and model construction. In our approach, a set of significant features was first selected from the experimental data using the RF-ACE tool. Then, a model was constructed using the given data.

Experimental data

In order to test our modeling methodology we examined a dataset produced in a study related to culture media optimization (unpublished data, Rahul Mangayil et al.). There, an enriched mixed microbial consortium was used in the bioconversion of crude glycerol to hydrogen, and the process was optimized in serum bottles by optimization of media components. The concentrations of five media components (NH_4Cl, K_2HPO_4, KH_2PO_4, $MgCl_2.6H_2O$, and KCl) were varied with the help of statistical design of experiments (Plackett-Burman, steepest ascent, Box-Behnken), and the resulting hydrogen production was measured (in mol-H_2/mol-glycerol). The data was modeled using first and second order polynomials in multiple linear regression. This data containing 35 samples is a typical data set produced during bioprocess modeling and optimization. Multiple linear regression is a useful tool for modeling the data from individual designs of the study but other methods are needed in order to model the entire data set at once.

Visualization and validation of models

In order to provide an overview to the models and the experimental data, visual representations were produced for the regularized regression model and the random forest model. Since visualization of the high dimensional variable space (five dimensions in our case study) is not feasible, the variables are visualized pair-wise. The values of remaining variables (three) are set in their average values calculated from the data. In addition, each model is assessed with *leave-one-out* (LOO) cross validation technique which estimates the accuracy of the predictions in an independent dataset.

Results and discussion

In our case study, we used multiple linear regression, regularized regression and random forests to predict the yield of hydrogen production. The performance of each method is evaluated by original dataset as well as transformed dataset with pairwise interactions and quadratic forms. Therefore, the original dataset contains 5 variables while the transformed dataset contains 20 variables.

Yield prediction using multiple linear regression

Multiple linear regression is used with and without nonlinearly transformed predictor variables to model the response variable. Without the transformed predictors, i.e., the simple model, the estimated correlation value (using the LOO cross-validation) was 0.65. However, using the transformed polynomial model the estimate for correlation decreased to a very low value of 0.012 and resulted in an insignificant model. This is mainly due to the aforementioned shortcomings of the multiple linear regression. It basically over-fits the model to the training samples and thus produces less accurate estimates for unseen data samples. Table S1 lists the model coefficients for the transformed polynomial regression model [see Additional file 1]. It can be noted that zero entries have been inserted to remove linearly dependent observations.

Yield prediction using regularized regression

First, we evaluated the simple model without the transformed variables. In this case, the parameter λ for the regularized regression is chosen by both manual selection and proper cross validation. In other words, we wanted to see if the results improve by manually selecting the lambda value optimally for each LOO cross validation fold. Although this is not possible in practical applications, it may give insight on the efficiency of

parameter selection using cross-validation with small sample size, and on the general applicability of a linear model for our problem.

As a result, the LOO correlation estimate becomes 0.85 with manual selection instead of 0.60 using proper cross-validation. The large gap between optimal and estimated correlation is at least in part due to the inaccuracy of the cross-validation type error estimators with small sample size; see, e.g., [9].

In the case of transformed polynomial regression model, the estimated value for correlation was found to be 0.69 which is higher than the case of the simple model. This clearly indicates the non-linear behavior of the original dataset. Table S1 shows the resulting coefficients in the constructed model where regularization has forced 5 out of 21 coefficients to zero [see Additional file 1]. Although, the same number of non-zero coefficients were obtained from the multiple linear regression as well but the main difference is the regularized coefficients. That is, the non-zero coefficients from regularized regression were also shrunk towards zero. This results in generalized models with higher overall prediction accuracy [3]. The yield predictions are visualized in Figure 1 as a response surface. In addition, the significant variables for the model and their corresponding coefficients are listed in Table 1.

Yield prediction using random forests

The RF-ACE tool [5] is used to build the random forests model. In our experiment, the type of the forest, the number of trees in the forest, and the fraction of randomly drawn features per node split are set to "RF", 20, and 10, respectively. All other parameters were kept to their default values. The results indicated that all variables were significant in the model. The yield predictions of the constructed model are visualized in Figure 2. In the accuracy examination, the RF-ACE model resulted in correlation of 0.88 (using LOO cross-validation). The capability of modeling non-linear relationships is the primary reason for high prediction accuracy in the constructed model. On the other hand, the model provided correlation value of 0.91 if the variable transformations

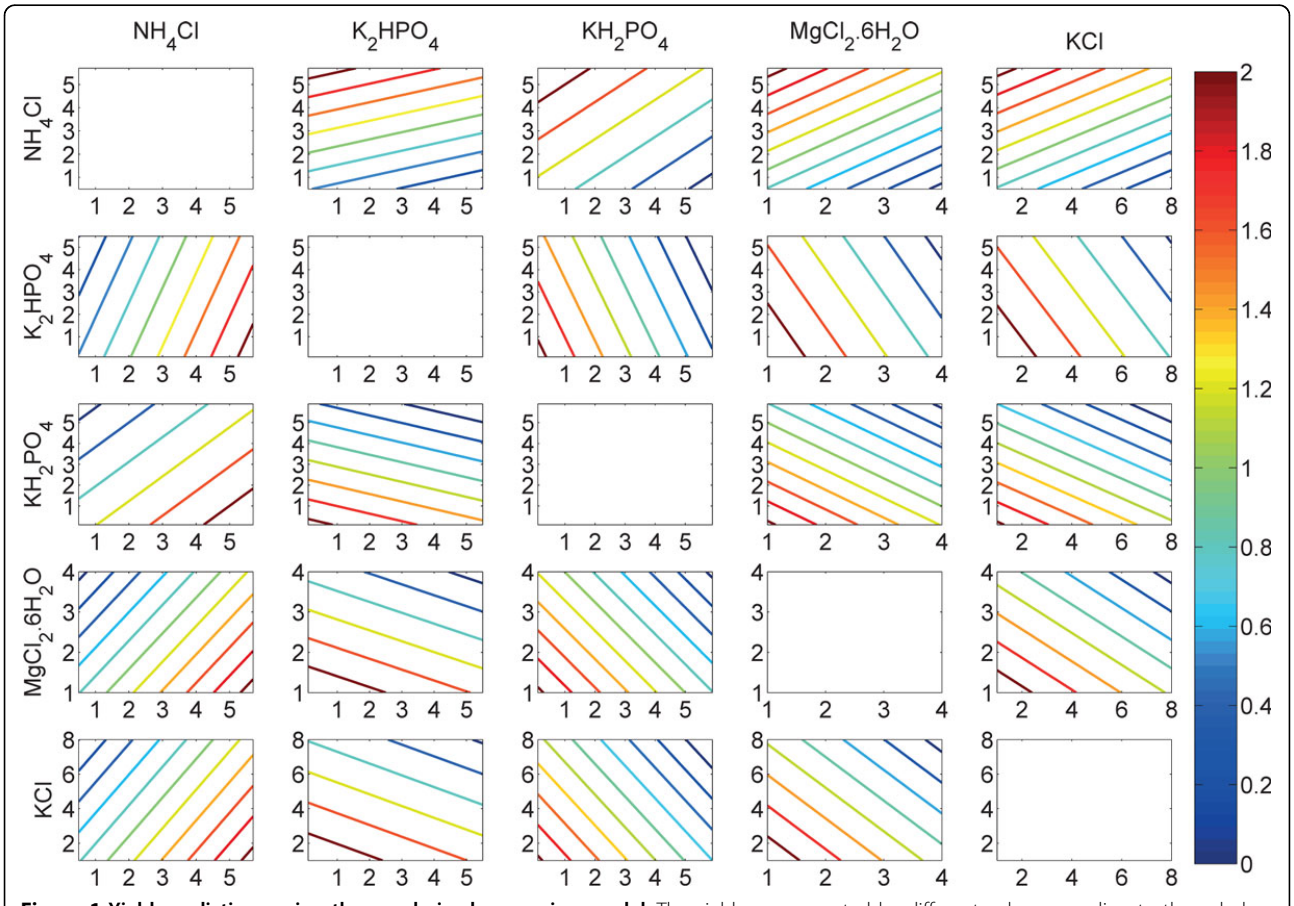


Figure 1 Yield predictions using the regularized regression model. The yields are presented by different colors according to the colorbar. The plots in the diagonal (i.e., variables are plotted against themselves) are left empty.

Table 1 Significant variables and their coefficients in the regularized regression model

Significant variables	Coefficient values
NH_4Cl	0.1254
K_2HPO_4	-0.0383
KH_2PO_4	-0.1061
$MgCl_2.6H_2O$	-0.1418
KCl	-0.0562

were used as additional predictor variables. Eventually, the increase is quite small, and may thus be a due to random fluctuation.

Method comparison

Both regularized regression with transformed variables and random forests produced results that are useful in bioprocess data mining. In particular, both methods determined all the variables significant and can be used to determine an advantageous control direction for them. The most notable difference in the results is the linearity that was in use in the regularized regression versus the nonlinearity that is inherent in random forests (see Figures 3 and 4). Simple linear models cannot fit to the nonlinearity of the data and, thus, the maximum response cannot be detected inside the examined space although it would be located in there. However, regularized linear regression with transformed variables was found successful in modeling the nonlinearity of the data to some extent. On the other hand, the random forest model is able to capture the nonlinearity. Here, the maximum response was determined approximately at the same point as in the media optimization study performed using the methods of statistical design of experiments.

Figure 3 and 4 show the performance of the three methods in yield prediction. It is clear that regularized linear regression failed to cope with data non-linearity unless transformed variables were used in regression. On the other hand, the use of transformed variables causes the multiple linear regression to fail. Thus, multiple linear regression is an efficient tool in the analysis of individual datasets designed by statistical design of experiments (e.g.,

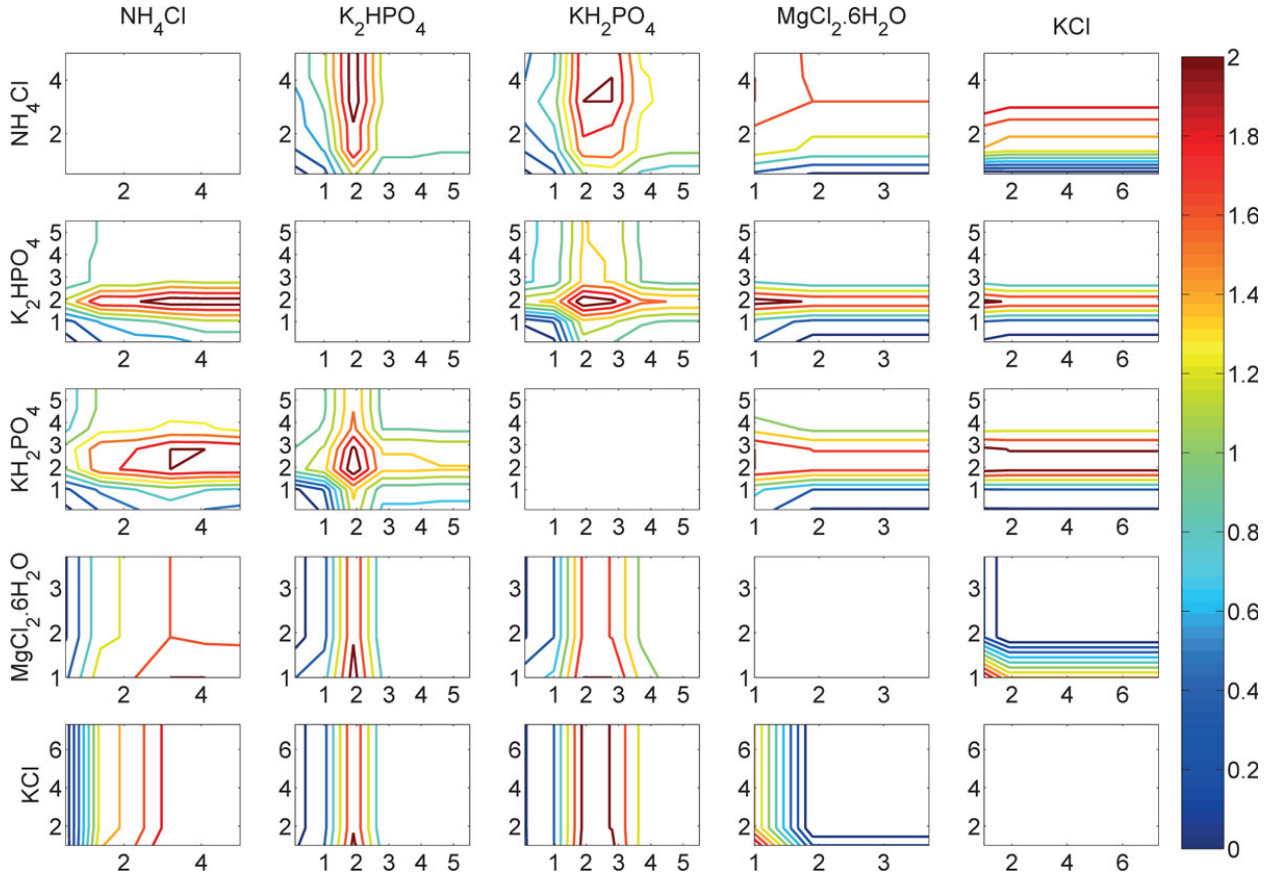


Figure 2 Yield predictions using the random forest model. The yields are presented by different colors according to the colorbar. The plots in the diagonal (i.e., variables are plotted against themselves) are left empty.

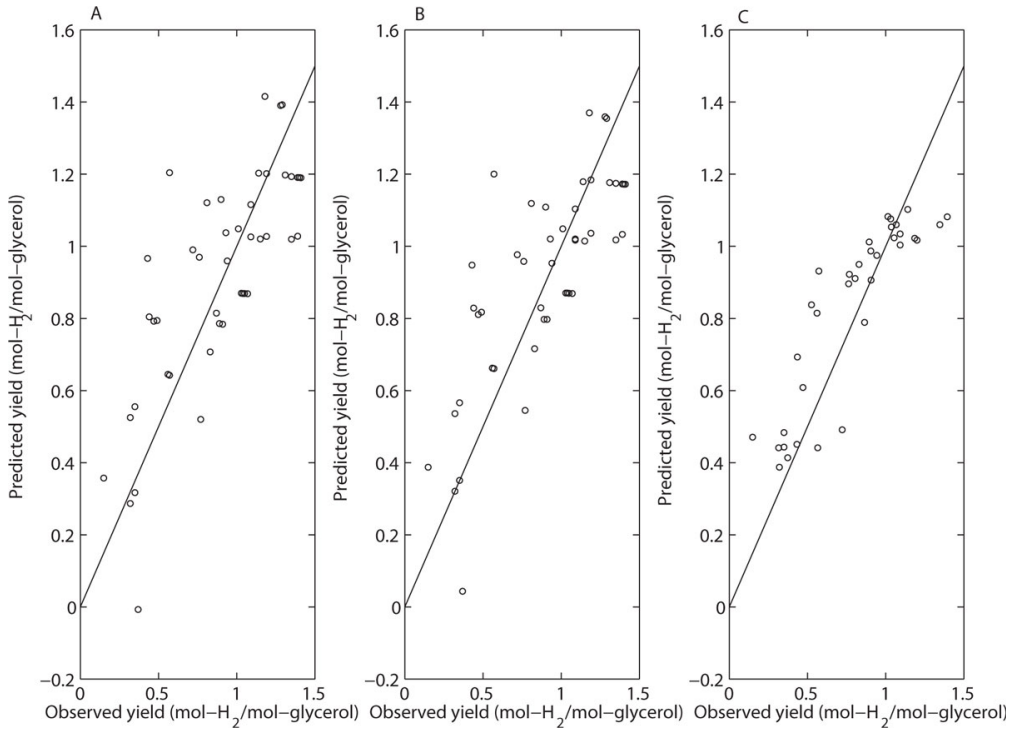


Figure 3 Comparison of prediction performance of models obtained by three methods for original dataset. **(A)** Multiple Linear Regression; **(B)** Lasso; **(C)** Random Forest. The straight line depicts perfect predictions should lie. The prediction accuracy for each model is estimated using LOO cross-validation.

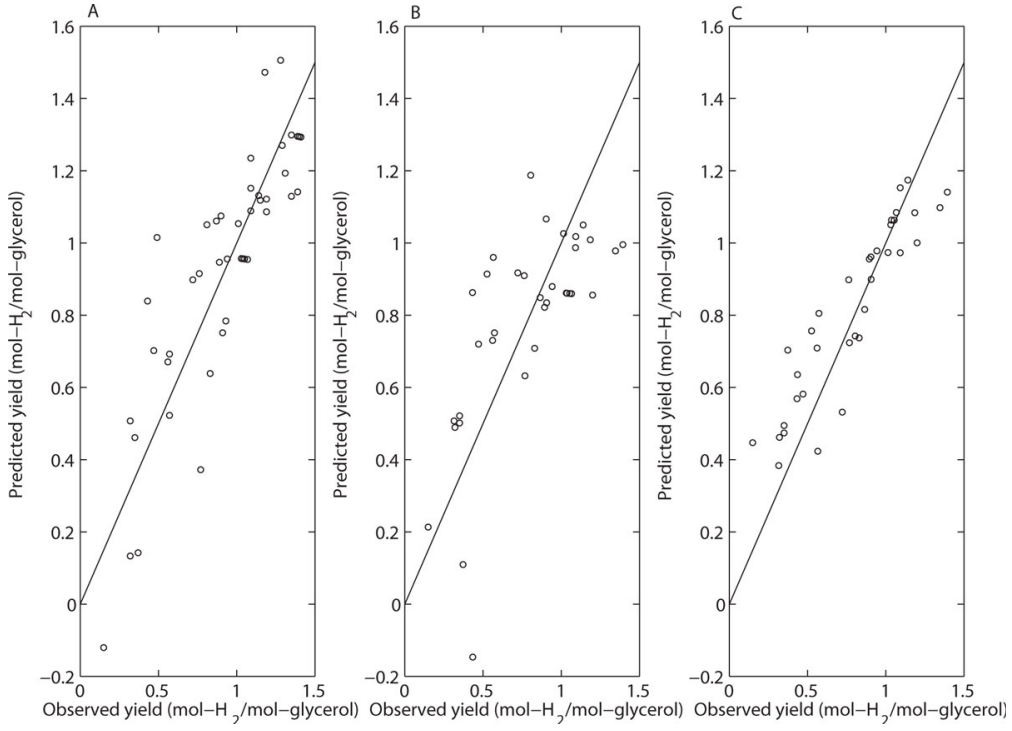


Figure 4 Comparison of prediction performance of models for the dataset containing the actual and the transformed variables. **(A)** Multiple Linear Regression; **(B)** Lasso; **(C)** Random Forest. The straight line depicts perfect predictions should lie. The prediction accuracy for each model is estimated using LOO cross-validation.

Plackett-Burman and Box-Behnken) but not useful in data mining of more complicated datasets like the one examined in here.

The LOO estimates for correlation ascertain that the RF-ACE provides a more accurate solution than the regularized regression. This, however, should not totally renounce the idea of using regularized regression as it mainly proves its worth in more complicated and high-dimensional data analysis. Moreover, linear regression has a useful feature of producing easily interpretable models and, on the other hand, the models are capable in producing predictions beyond the already examined parameter space.

Conclusions

In this study, we applied two novel data analysis methods (regularized regression and random forests) in bioprocess data mining and compared them to multiple linear regression that is commonly applied in relation to statistical design of experiments. Both of the studied methods were able to produce models that fit to the examined data. In particular, the non-linearity of the data was well modeled by random forests. This property is very valuable in data mining of multiple integrated data sets. As the results demonstrated, traditionally used multiple linear regression does not perform satisfactorily in non-linear input-output relations. The traditional approach using the first and the second order polynomial models would face further problems if the data was multimodal. In the future, it would be of interest to further study regularized regression and random forests in bioprocess data mining. This could mean, for example, the inclusion of categorical variables in the data and studies with different types of bioprocesses.

Additional material

Additional file 1: as PDF - Table S1: Significant coefficient values in different methods using transformed data. This file contains a table describing the coefficient values generated by Lasso and multiple linear regression methods for the transformed dataset. Here, the coefficient β_0 represents the intercept, β_1 corresponds to variable NH_4Cl, β_2 to K_2HPO_4, β_3 to KH_2PO_4, β_4 to $MgCl_2.6H_2O$ and β_5 to KCl, respectively.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

SSH and MF made substantial contribution in writing the manuscript, interpretation of the data, and design and analysis of the models. RM was responsible in acquisition of the data. TA and HH contributed to the design of the study, and in writing and revising the manuscript.

Acknowledgements

The authors thank the Academy of Finland, project "Butanol from Sustainable Sources" (decision number 140018), for funding the study.

Declarations

The publication cost for this work was supported by the Academy of Finland, project "Butanol from Sustainable Sources" (decision number 140018).

This article has been published as part of BMC Systems Biology Volume 7 Supplement 1, 2013: Selected articles from the 10th International Workshop on Computational Systems Biology (WCSB) 2013: Systems Biology. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcsystbiol/supplements/7/S1.

Authors' details

[1]Department of Signal Processing, Tampere University of Technology, Tampere, P.O. Box 553, 33101, Finland. [2]Department of Chemistry and Bioengineering, Tampere University of Technology, Tampere, P.O. Box 541, 33101, Finland.

Published: 12 August 2013

References

1. CDER: **Process Validation: General Principles and Practices.** 2011 [http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/ucm070336.pdf].
2. Tuv E, Borisov A, Runger G, Torkkola K: **Feature selection with ensembles, artificial variables, and redundancy elimination.** *Journal of Machine Learning Research* 2009, **10**:1341-1366.
3. Tibshirani R: **Regression shrinkage and selection via the Lasso.** *J R Statist Soc B* 1996, **58(1)**:267-288.
4. Mathworks: *Matlab* Natick, MA; 2011.
5. **RF-ACE: multivariate machine learning with heterogeneous data.** [http://code.google.com/p/rf-ace/].
6. Andersen PK, Skovgaard LT: **Multiple regression, the linear predictor.** In *regression with linear prediction. Volume 0.* New York, NY: Springer; 2010:231-302.
7. Miller AJ: **Subset Selection in Regression.** Chapman and Hall/CRC;, 2 2002.
8. Breiman L: **Random forests.** *Machine Learning* 2001, **45(1)**:5-32.
9. Saeys Y, Inza I, Larranaga P: **A review of feature selection techniques in bioinformatics.** *Bioinformatics* 2007, **23(19)**:2507-2517.

doi:10.1186/1752-0509-7-S1-S5
Cite this article as: Hassan *et al.*: Bioprocess data mining using regularized regression and random forests. *BMC Systems Biology* 2013 **7**(Suppl 1):S5.

Publication VI

M. Farhan, A. Larjo, O. Yli-Harja, and T. Aho, "Modeling bioprocess scale-up utilizing regularized linear and logistic regression," in *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, Southampton, UK, September 22-25, 2013, pp. 1-6

2013 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2013, SOUTHAPMTON, UK

MODELING BIOPROCESS SCALE-UP UTILIZING REGULARIZED LINEAR AND LOGISTIC REGRESSION

Muhammad Farhan[1], *Antti Larjo*[1,2], *Olli Yli-Harja*[1], *Tommi Aho*[1*]

[1]Tampere University of Technology, P.O. Box 553, FI-33101 Tampere, Finland
[2]Aalto University School of Science, 15400, FI-00076 Aalto, Finland
muhammad.farhan@tut.fi, antti.larjo@tut.fi, olli.yli-harja@tut.fi, tommi.aho@tut.fi

ABSTRACT

Bioprocess scale-up from optimized flask cultivations to large industrial fermentations carries technical challenges and economical risks. Essentially, the prediction of optimal process conditions in large fermentations based on small scale experiments is non-trivial. For example, common statistical methods encounter problems with the high-dimensional, small sample size and, on the other hand, the use of various scale-up criteria requires *a priori* knowledge that may be difficult to obtain. We propose a novel computational scale-up approach applicable to various bioprocesses. The method bases on regularized linear and logistic regression. With embedded feature selection, it automatically identifies the most influential parameters and predicts their values in large scale. In addition, the method predicts the resulting large scale yield. As a case study, we examined the production of a cytotoxic compound. We predicted scale-up from flask and 2L to 30L fermentations and found that, in both cases, the product yield predictions are close to experimentally observed yields.

Index Terms— Bioprocesses modeling, scale-up, regression model, yield prediction, regularized logistic regression.

1. INTRODUCTION

Today, fermentation bioprocesses produce many of the active pharmaceutical ingredients, enzymes and fine chemicals. Because of the high costs in large industrial scale, the process optimization is performed in laboratories at smaller scales, for example in flasks. An important step in bioprocess development is the process scale-up where the production obtained in laboratory size equipment (for example, flasks) is scaled up to industrial size (for example, 1000 liter fermenters).

During the last decades, extensive research has been performed [1, 2] for identifying applicable scale-up strategies that would result at least the same product yield in large scale that has been obtained in the preceding small scale process optimization. However, it is generally acknowledged that there does not exist only one strategy that is applicable for all process types but the strategy depends on the process characteristics and the produced product [2]. Commonly, the search for a suitable scale-up strategy starts by characterizing the key stress factors and parameters influencing cell growth and product yield. Then, the process is optimized in small scale, and a so-called scale-up criterion is established. Scale-up criteria are different kinds of conversions of specific operational parameters into criterion values that are maintained constant across scales. The criteria suggested in the literature include, for example, volumetric mass transfer coefficient k_La, power per unit volume, concentration of dissolved oxygen, impeller tip speed, pH change of the medium, and mixing time [3, 4, 5, 6, 7]. When applying scale-up criteria, the process developer assumes that cell growth and product yield remain constant if the selected criterion value is kept constant across scales. This makes it possible to determine the values of particular operational parameters in different scales. However, the use of scale-up criteria is able to provide only a partial solution as they can be used to determine the values of few parameters only while a typical bioprocess involves tens of parameters.

Apart from traditional approach of defining criteria, the scale-up task has also been approached by the methods of statistical modeling as conceived by the authors of [8, 9, 10]. Response surface methodology (RSM) is efficient in optimization of several variables as well as studying interactions between them. In [10], RSM is used to optimize the production within a single scale and to help in scale-up of extracellular protease from *Bacillus* sp. The authors of [8] suggest an RSM-based methodology for examining whether a single parameter appears as an issue in scale-up. However, neither of these methods can be used to predict the values of operational parameters in large scale based on the samples in small scale.

Here, we aim at preparing a scale-up model that makes it possible to determine the values of operational parameters in large scale fermenters given the values of operational parame-

*The authors thank Dr. Heikki Huttunen for valuable comments and Galilaeus Oy for providing their experimental data. Financial support from Finnish Programme for Centre of Excellence in Research 2006-2011, the Academy of Finland, Tekes - the Finnish Funding Agency for Technology and Innovation and from Nokia Foundation is also acknowledged.

ters in small scale cultivations or fermentations. The model is required to scale-up the given small scale sample such that the product yield in the predicted large scale sample and the given small scale sample are roughly the same. The prediction task can be formulated into a statistical problem which has the following challenges: First, typical problems in bioprocess modeling have a small number of samples with respect to the number of parameters (that is, the problems are of type Large P, Small N). Second, many operational parameters obtain categorical values and their analysis is difficult using conventional methods. Third, different equipment have different parameter types, and their values are not directly comparable (e.g., flask shaking vs. fermenter mixing). Fourth, the processes exhibit nonlinear behavior that is difficult to capture. The proposed method copes with all these challenges using the state-of-the-art methods of statistical modeling. The advantage of using the proposed approach over the approaches based on constant criteria, like k_La, is that the proposed approach identifies the relationships between various operational parameters, as well as their effects on the process outcome. Therefore, unlike when using one of the constant criteria, the proposed method does not need *a priori* knowledge in criterion selection but the method automatically selects the most important variables from a large set of variables. Consequently, the final model only contains variables that have effect on the process outcome.

The rest of the paper is organized as follows. Section 2 describes the proposed methodology for scale-up. Section 3 describes experimentation and discusses the obtained results. Finally, Section 4 concludes the paper.

2. SCALE-UP METHODOLOGY

In this section, we describe our scale-up methodology. It consists of four steps: encoding of categorical variables to incorporate them into modeling, product yield prediction using regularized linear regression, yield correspondence-based data rearrangement and scale-up modeling using regularized linear and logistic regression. Fig. 1 highlights the procedure for scale-up model development and its testing.

A multiple regression model [11] is a model with multiple, p, independent variables $x_{i1}, x_{i2}, \ldots, x_{ip}$ per sample i and involves $p+1$ regression coefficients $b_0, b_1, \ldots, b_p$. The model is called multiple linear regression model when these coefficients are linearly related to each other to predict the dependent variable y_i given by the expression

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \ldots + b_p x_{ip}, \tag{1}$$

equivalently written in matrix form for i = 1, 2, 3, …, N as

$$\mathbf{y} = \mathbf{H}\boldsymbol{\theta} = [\mathbf{1X}]\boldsymbol{\theta}, \tag{2}$$

where $\mathbf{y} = [y_1 y_2 \ldots y_N]^T, \mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_N]^T, \mathbf{x}_i = [x_{i1} x_{i2} \ldots x_{ip}]^T$, and $\boldsymbol{\theta} = [b_0 b_1 b_2 \ldots b_p]^T$. Estimation of the coefficients $\hat{\boldsymbol{\theta}}$ is performed such that the sum of squared error in the observed and predicted values of y_i is minimized.

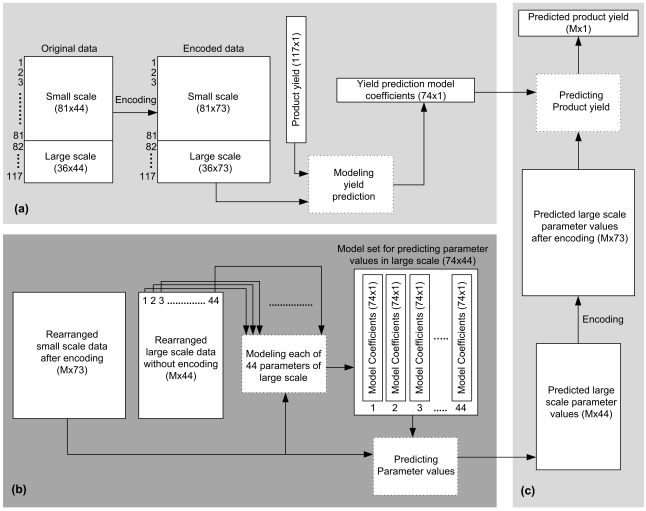


Fig. 1. Methodology for scale-up modeling and testing. Procedure for (a) modeling yield prediction in all scales, (b) modeling operational parameter values in large scale using the data at a smaller scale and (c) testing the developed models.

In the light of above modeling scheme, the general problem statement that can be defined for the scale-up is to

$$\text{find } \hat{\boldsymbol{\theta}}_j : \mathbf{x}_{Lj} = [\mathbf{1X}_S]\boldsymbol{\theta}_j \text{ given that } |\mathbf{y}_L - \mathbf{y}_S| \leq \epsilon \tag{3}$$

where $j = 1, 2, \ldots, p$ and which results in

$$\hat{\mathbf{X}}_L = [\hat{\mathbf{x}}_{L1} \hat{\mathbf{x}}_{L2} \ldots \hat{\mathbf{x}}_{Lp}], \text{ such that}$$
$$\hat{\mathbf{y}}_L = [\mathbf{1}\hat{\mathbf{X}}_L]\hat{\boldsymbol{\theta}}^{yieldprediction} \approx \mathbf{y}_L, \tag{4}$$

where the subscripts L and S are used for large and small scale data respectively and all values of $\hat{\boldsymbol{\theta}}$ are determined by using regularized linear and logistic regression for numerical and categorical variables respectively. The value of allowed difference ϵ can be chosen. In our case study (described in Section 3), ϵ was set to 0.2, and the values of the product yields in the experiments were normalized to the range (0, 1] in order to anonymize the data.

2.1. Encoding of categorical variables

Bioprocesses often involve categorical variables which hold categorical labels rather than numerical values. Categorical variables with two labels are called dichotomous variables and can be directly entered as predictor or predicted variables in a multiple linear regression model [11]. In such a case using the expression in (1) only requires that the labels in a categorical variable be replaced with binary code like 0 and 1. However, in most cases the categorical predictors hold more than two labels which cannot be directly incorporated into the regression model. They require some other coding or transformation in order to be incorporated in the regression analysis. One way is to code a categorical variable with k labels

into $k-1$ dichotomous variables. For example, if a categorical variable has six labels then five dichotomous variables could be constructed that would contain the same information as the original variable, see the matrix c_1 at the bottom. This process of encoding a categorical variable into a number of separate, dichotomous variables is called dummy coding [11].

There exist many different ways or coding systems to dummy code a categorical variable. The coding system should be chosen so that it highlights the comparisons that are meant to be done among the different labels. Moreover, if there was high correlation or linear dependency among the variables, the regression modeling would become inaccurate. Therefore, the appropriate coding system and the regression methodology should minimize correlation and linear dependence. Moreover, often the processes data consists of samples for which the values of specific categorical variables for a particular set of measurements may not vary significantly (many experiments are performed with one or two changed parameters). In such cases, using binary coding is not a proper choice since it may cause singularity issues because of giving the same binary value to every label of all the categorical variables. Instead, unique codes and proper contrasts are required for each categorical variable.

One of the feasible coding systems is called contrast coding in which the labels are coded in such a manner that creates contrast among a set of labels [11]. The contrast is typically produced by giving the same variable positive and negative values for the labels between which the contrast is meant to be created. An example of contrast coding based dummy coding of a categorical variable with six labels is given in the matrix c_2 below. Here, the first new variable creates contrast between the groups of first two and the rest of the labels. The second new variable creates contrast between the first two labels. The third new variable creates contrast between groups of label 3, label 4 and of label 5, label 6 and so on. The advantage of defining such codes is that even if the labels for different variables remain unchanged for some samples, the values held by the subsequently created dummy coded variables would not only be different but also have contrast among the labels. Using contrast values helps in finding such coefficients of the model which can later be used to correctly distinguish between the categorical labels when the coded values are presented. Therefore, in the first step, fixed length dummy codes are defined in this way for each of the categorical variables while the numerical variables remain unchanged.

$$c_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, c_2 = \begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & -1 & 0 \\ 1 & 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 & -1 \end{bmatrix}$$

2.2. Product Yield Modeling

Once the original data is translated into a form that can be used for modeling, that is, after dummy coding, a generalized product yield prediction model is needed, like (2) where $\mathbf{x}_i$ takes the sample values and y_i takes the values of the product yield for experiment sample i. The resulting model coefficients can be used to predict the product yield for a given measurement sample at any scale, using the model presented in (2). The significance of developing such a model is that it is needed at a later stage when the large scale sample parameter values are predicted using the small scale samples, and it is desired to predict the product yield for that scaled-up sample to check how accurate the achieved scale-up is. Fig. 1(a) illustrates the concept of yield prediction modeling.

The problem with such process modeling is that they contain tens of different experimental variables and the number almost gets doubled after dummy coding. Not all of the variables are essential for modeling, and also the complexity of the model increases with the amount of variables used in the regression modeling. Moreover, incorporating all the variables in regression may lead to over-fitting. Therefore, the selection of the best subset of variables for model development, that is, feature selection, is required. Regularized regression with embedded feature selection has been found to be very effective in such situations [12, 13]. It is normally used when there are more variables than measurements, that is $P \gg N$, or the variables are linearly dependent on each other or overfitting is prohibiting the generalization of the solution. Here, we propose to use the regularized regression method of Least Absolute Shrinkage and Selection Operator (LASSO) that penalizes on the coefficients magnitude by adding a penalty term to the prediction error. That term includes a constant factor λ by which coefficients are translated to shrink them towards zero as well as towards each other [12]. Therefore, it always gives sparse solution, that is, many of the coefficients become zeros. This is how it automatically incorporates feature and model selection into optimization [14] as only the best variables, corresponding to non-zero or significant coefficients, are selected and the model is simplified.

Here we have N measurement samples forming the predictor variables vector $\mathbf{X} \in \mathbb{R}^{N \times P}$ and predicted or response variable vector $\mathbf{y} \in \mathbb{R}^{+}$. Assuming that inputs x_{ij} is standardized, that is, it has zero mean and unit norm, if the linear regression model is similar to (1), then the estimate of the model coefficients provided by the shrinkage method of LASSO is given by

$$\hat{\boldsymbol{\theta}} = \arg\min_{\theta} \sum_{i=1}^{N} (y_i - b_0 - \sum_{j=1}^{p} x_{ij} b_j)^2 \qquad (5)$$
$$\text{subject to} \sum_{j=1}^{p} |b_j| \leq t,$$

which by using (1) and (2) is equivalent to minimizing the prediction error-based Lagrange function given by

$$\|\mathbf{y} - \mathbf{H}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1, \qquad (6)$$

where $\lambda > 0$ is the Lagrange multiplier also called regularization parameter which controls the amount of shrinkage of the

coefficients (at some large value of λ, all coefficients are zero) and $\|\boldsymbol{\theta}\|_1$ is the l_1-norm of the coefficient vector by which the error function is penalized. The solution in (5) is non-linear in y_i, and therefore there is no closed form expression of the problem but it may be solved using quadratic programming or the *Least Angle Regression* algorithm [15, 12]. It gives solutions for different values of λ and choosing the optimal solution from them is non-trivial. A graph called regularization path visualizes the values of the coefficients for all the values of $\lambda > 0$ and may help in finding the optimal solution. However, in practice, cross-validation is usually performed over a set of values of λ to estimate the prediction error and to pick the optimal solution corresponding to the minimum prediction error. Here, we use 10-fold cross-validation to select the optimal model coefficients and also to ensure that the model is general enough to give a very low prediction error for the product yield even for an unseen sample.

2.3. Yield Correspondence-based Data Rearrangement

Since the aim in the process development is to optimize the process in smaller scale and to preserve the product yield in the large scales, the product yield can be used as a reference for correspondence-based data rearrangement. The idea is that if a pair of samples at both the scales have the product yield within a specific range then there exist a one-to-one correspondence between the sample pair. Since exactly matching product yield values is highly unlikely, so there has to be some tolerance band for the product yield correspondence. As stated in (3), here we use $\pm$ 0.2 units tolerance for the difference in product yield. Each sample of large scale is used to find the corresponding sample(s) from the small scale such that the difference in the product yield is within $\pm$ 0.2 units. One large scale sample can have more than one corresponding small scale samples. In that case, the large scale sample is replicated as many times as there are corresponding small scale samples. Therefore, after rearrangement, the new data contains equal number of samples for both the scales.

2.4. Scale-up Modeling

Once the data rearrangement is performed, the aim is to develop optimal linear models to predict the variable values in the large scale based on the variable values in the small scale (see (3)). In the previous section, we discussed how regularized linear regression can produce sparse models, however, the response variable in that case was numerical whereas the operational parameters are categorical as well. If the response variable is categorical, that regression modeling technique cannot be used alone to develop a model for prediction or classification of its labels. Here we exploit sparse logistic regression, a framework of LASSO [16, 17] to solve this problem.

The essence of logistic regression is the logistic function which is used to model the posterior probability density function (PDF) for each class or label. These class probability densities are then used to define the classifier. The PDF for the class $k = 1, 2, ..., K$ is modeled as

$$p_k(\mathbf{x}) = \exp(\boldsymbol{\theta}_k^T \mathbf{x})/(1 + \sum_{j=1}^{K} \exp(\boldsymbol{\theta}_j^T \mathbf{x})), for k \neq K, \quad (7)$$

$$\text{and } p_K(\mathbf{x}) = 1/(1 + \sum_{j=1}^{K} \exp(\boldsymbol{\theta}_j^T \mathbf{x})), \quad (8)$$

where $\mathbf{x} = [1 x_1 x_2 ... x_p]^T$ denotes the augmented predictor vector and $\boldsymbol{\theta}_k = [b_{k0} b_{k1} b_{k2} \ldots b_{kp}]^T$ are k set of coefficients of the models, one for each of the k categorical label, and are obtained by maximizing the penalized log-likelihood given by

$$\hat{\boldsymbol{\theta}}_{1,2,...,K} = \underset{\boldsymbol{\theta}_{1,2,...,K}}{\arg\max} [\frac{1}{N} \sum_{i=1}^{N} \log p(x_i) - \lambda \sum_{j=1}^{K} \|\boldsymbol{\theta}_j\|_1], \quad (9)$$

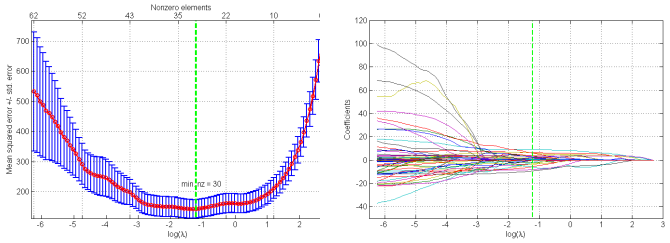
where $\boldsymbol{\theta}_{1,2,...,K} \in \mathbb{R}^{(p+1)\times K}$ and whose quadratic approximation gives rise to an equivalent penalized iteratively reweighted least squares problem that can easily be solved by coordinate descent algorithm [16]. Again cross-validation governs the selection of the optimal model coefficients such that the prediction error is minimal. When a measurement sample is presented and its corresponding class label is to be predicted, the model coefficients and the predictor values are used to compute the probability densities for every class labels using (7) and (8). The class with the highest probability is the predicted class label for the categorical variable.

Hence regularized linear regression is used to develop models for numerical variables whereas logistic regression framework of LASSO is used for modeling categorical variables. Scale-up is realized by using these models in predicting the value of every individual variable at large scale. This concept is illustrated in Fig. 1(b).

3. EXPERIMENTAL RESULTS AND DISCUSSION

Materials: Our case study contains data about 117 samples from a bioprocess that produces a cytotoxic compound called anthracycline. Experiments were performed in flasks (81 samples), 2L fermenters (24 samples), and 30L fermenters (12 samples). Since the experiments at 30L fermenters are expensive to perform, the process optimization has been performed in flasks and 2L fermenters and the number of samples in 30L is much smaller. This typical situation highlights the need of developing efficient scale-up modeling approaches. Researchers should be able to complete the process optimization in small scale, and have a model that predicts the values of operational parameters in the large scale. However, in order to develop a general model, the required minimum number of samples in each scale is difficult to determine and usually remains unknown.

Testing the Product Yield Modeling: Each sample is composed of around 40 typical bioprocess variables such as strain,



(a) (b)

Fig. 2. Product yield prediction model selection (Green dashed vertical line) using the (a) error plot and (b) regularization path plotted as a function of $log(\lambda)$.

broth medium, broth adsorbent, fermentation time, temperature, agitation, aeration, pH, etc. After dummy coding the categorical variables, the number of variables increased to more than 70. All the samples were exploited when developing the yield prediction model using regularized linear regression. Here, 10-fold cross-validation was performed to determine the optimal model coefficients based on the minimum prediction error so that the model is general enough to predict the product yield for an unseen sample. Fig. 2(a) plots the mean square error (MSE) with the standard error obtained from cross-validation as a function of logarithm of the regularization parameter λ. The vertical green dashed line is at the point with the minimum error. Fig. 2(b) plots the regularization path as a function of $\log(\lambda)$. The value of λ yielding minimum prediction error gives the optimal solution (coefficients at $\log(\lambda)$ = -1.2127 in regularization path) with 43 out of 73 model coefficients as zero. Half of the 30 non-zero coefficients (corresponding to dummy-coded variables) appear significant, and only around 15 real variables remain with a true impact on the product yield. This highlights the capability of our regularized regression-based approach for automatically selecting the important variables from a large set of variables. A benefit of our approach is that no biological *a priori* knowledge is needed about the variables or their relationship with the product yield. Fig. 3 compares the experimentally observed and the predicted values of the product yield. It is evident that despite of small sample size, categorical variables and high-dimensionality of the data, the methodology identifies a model that is both general and accurate.

Testing the Scale-up Modeling: In order to test the scale-up modeling, samples with specific product yields were selected from the small scale data to predict the variable values of their corresponding large scale samples. As mentioned earlier, the value for every individual variable of a large scale sample was predicted using the respective model. That is, the values of numerical variables were predicted using (2) whereas the categorical variables were predicted using the PDF models of (7) and (8). Then the predicted variable values were given to product yield prediction model that predicted the product yield in large scale. Since the aim was that product yield remains constant in scale-up, this predicted value should be within $\pm$ 0.2 units tolerance range of the product yield at the small scale. Fig. 1, in particular Fig. 1(c), shows how the testing of scale-up is performed.

The objective in our case study is to use the flask and 2L experiments as two alternative small scale data and to develop models that are able to determine the values of operational parameter at 30L. Moreover, we aimed to determine whether the scale-up to a 30L fermenter is possible directly from the flask scale, or is it the only option to base the scale-up on fermentations at 2L scale. If direct scale-up from flask to 30L is possible, that would improve the cost-efficiency in process development. Therefore, flask experiment data was first used as the small scale data and 30L data as the large scale data. This setting produced 330 measurement sample pairs after data rearrangement. Then, the values of operational parameters in 30L experiments were predicted using the small scale samples. The derived large scale samples were then provided to the product yield prediction model to predict the product yields. The performance of the scale-up strategy was evaluated by comparing the predicted product yields with the experimental product yields at the large scale. Fig. 4(a) shows the comparison of the experimental and the predicted product yields at the large scale. In general, the values of the product yield using the derived large scale samples are very much in accordance with the $\pm$ 0.2 units tolerance range of the experimental large scale product yield. This is also ascertained by the root-mean-square error (RMSE) of the product yield which is 0.172 and is within the tolerance range of 0.2.

The total prediction error of the product yield is composed of two different errors sources: the errors in predicting the parameter values at the large scale, and the error caused by the yield prediction model. The small total prediction error of our approach suggests that the scale-up is not only good in terms of predicted product yield but also in the sense that the derived large scale samples are quite similar to the experimentally tested large scale samples. That is, there is only little difference in the values of operational parameters in the predicted large scale samples and the experimental large scale samples. In particular, this is true for the 15 operational parameters that have significant effect to the product yield.

In the second step in our case study, samples from the ex-

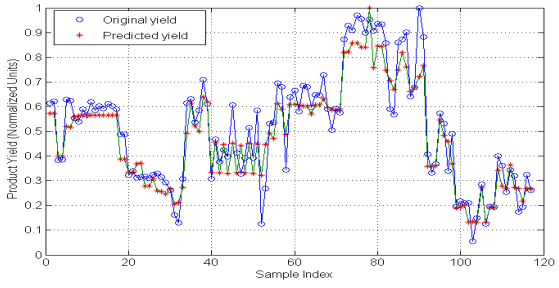
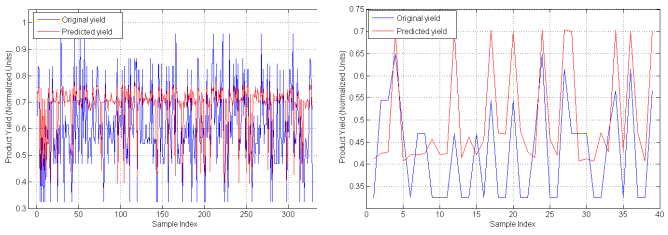


Fig. 3. Result of product yield (normalized to the range (0, 1]) prediction model for the samples from flask experiments compared to the experimentally observed product yields.

periments at 2L and 30L fermenter were considered as the small scale and large scale data, respectively. After applying the same procedure described in the previous paragraph we derived large scale (30L) samples and predicted their product yield. The comparison of the experimental product yield at large scale with the predicted product yield is presented in Fig. 4(b). It shows that the scale-up from 2L to 30L is satisfactory since in most of the cases the product yield from the scaled-up samples is within the $\pm$ 0.2 units tolerance range. Again, this is confirmed by the RMSE value of the product yield which is 0.14 and is within the tolerance range of 0.2. Thus, in this case study, our modeling methodology was able to achieve the scale-up with very similar accuracy from flask to 30L and from 2L to 30L. Finally, with the results of Fig. 4 we can infer that it is appropriate to perform 30L fermentations directly after flask experiments rather than to perform experiments at 2L vessels too. Therefore, in future we may consider omitting fermentations at the 2L scale.

4. CONCLUSION

A novel statistical approach to model the scale-up of bioprocesses was presented. Regularized regression with the embedded feature selection property of LASSO and its logistic regression classification framework provided effective tools for modeling the scale-up. The approach contained two modeling tasks. First, a model was developed to predict the product yield scale-independently. The model was found to be general enough and providing satisfactory results in different scales. Second, the scale-up modeling (i.e., the prediction of operational parameters) was realized by developing a separate model for each parameter in the large scale such that the value of the parameter is predicted based on the operational parameters in a small scale. Illustrations revealed that the scale-up was successfully achieved from flask to 30L fermenter, as well as from 2L to 30L fermenter. This was ascertained by the RMSE values that were well within the specified range of 0.2. Because of similar performance in these two cases, the idea of omitting the 2L fermentations in future is supported. Instead of these lab-size fermentations, the values of operational parameters in 30L fermenters could be determined based on flask experiments. The future work contains more detailed characterization of the presented methodology, for example, by testing with different experimental data with various production organisms and products.



(a) (b)

Fig. 4. The predicted product yield (normalized to the range (0, 1]) after scale-up to 30L experiment from (a) flask experiment and (b) 2L experiment, compared to the experimentally observed product yield of 30L experiment samples.

5. REFERENCES

[1] Junker HB, "Scale-up methodologies for escherichia coli and yeast fermentation processes," *Bioscience and Bioengineering*, vol. 97, pp. 347–364, 2004.

[2] Schmidt FR, "Optimization and scale up of industrial fermentation processes," *Applied Microbiology and Biotechnology*, vol. 68, pp. 425–435, 2005.

[3] Garcia-Ochoa F and Gomez E, "Bioreactor scale-up and oxygen transfer rate in microbial processes: An overview," *Biotechnology Advances*, vol. 27, pp. 153–176, 2009.

[4] Katzer W, Blackburn M, Charman K, Martin S, Penn J, and Wrigley S, "Scale-up of filamentous organisms from tubes and shake-flasks into stirred vessels," *Biochemical Engineering*, vol. 7, pp. 127–134, 2001.

[5] Marques MPC, Cabral JMS, and Fernandes P, "Bioprocess scale-up: quest for the parameters to be used as criterion to move from microreactors to lab-scale," *Chemical Technology and Biotechnology*, vol. 85, pp. 1184–1198, 2010.

[6] Rocha-Valadez JA, Estrada M, Galindo E, and Serrano-Carreon L, "From shake flasks to stirred fermenters: Scale-up of an extractive fermentation process for 6-pentyl-a-pyrone production by trichoderma harzianum using volumetric power input," *Process Biochemistry*, vol. 41, pp. 1347–1352, 2006.

[7] Seletzky JM, Noak U, Fricke J, Welk E, Eberhard W, and Knocke C et al, "Scale-up from shake flasks to fermenters in batch and continuous mode with corynebacterium glutamicum on lactic acid based on oxygen transfer and ph," *Biotechnology and Bioengineering*, vol. 98, pp. 800–811, 2007.

[8] Hsu Y-L and Wu W-T, "A novel approach for scaling-up a fermentation system," *Biochemical Engineering*, vol. 11, pp. 123–130, 2002.

[9] Ogawa S, Kamijima T, Miyamotao Y, Miyajima M, Sato H, and Takayama K et al, "A new attempt to solve the scale-up problem for granulation using response surface methodology," *Pharmaceutical Sciences*, vol. 83, pp. 439–443, 1994.

[10] Saran S, Isar J, and Saxena RK, "Statistical optimization of conditions for protease production from bacillus sp. and its scale-up in a bioreactor," *Applied Biochemistry and Biotechnology*, vol. 141, pp. 229–240, 2007.

[11] Stockburger DW, *Multivariate statistics: concepts, models, and applications*, Missouri State University, 2001.

[12] Hastie T, Tibshirani R, and Friedman JH, *The elements of statistical learning: Data mining, inference, and prediction*, Springer Series in Statistics, Springer, 2009.

[13] Kauppi J-P, Huttunen H, Korkala H, Jääskeläinen IP, Sams M, and Tohka J, "Face prediction from fmri data during movie stimulus: Strategies for feature selection," in *Proc. International Conference on Artificial Neural Networks ICANN*, 2011, pp. 189–196.

[14] Tibshirani R, "Regression shrinkage and selection via the lasso," *Royal Statistical Society Series B Methodological*, vol. 58, pp. 267–288, 1996.

[15] Efron B, Hastie T, Johnstone I, and Tibshirani R, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[16] Friedman JH, Hastie T, and Tibshirani R, "Regularization paths for generalized linear models via coordinate descent," *Statistical Software*, vol. 33, pp. 1–22, 2010.

[17] Huttunen H, Kauppi J-P, and Tohka J, "Regularized logistic regression for mind reading with parallel validation," in *Winning submission to Mind reading from MEG, PASCAL Challenge in ICANN*, 2011, pp. 20–24.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3274-0
ISSN 1459-2045